Neural Processing Letters (2022) 54:4495–4519
https://doi.org/10.1007/s11063-022-10819-4

# BARLAT: A Nearly Unsupervised Approach for Aspect Category Detection

**Avinash Kumar[1] · Pranjal Gupta[1] · Nisarg Kotak[1] · Raghunathan Balan[1] · Lalita Bhanu Murthy Neti[1,2] · Aruna Malapati[1]**

Accepted: 4 April 2022 / Published online: 23 April 2022


**Abstract**
Aspect category detection is an essential task in aspect-based sentiment analysis. Most previous works use labeled data and apply a supervised learning approach to detect the aspect category. However, to avoid the dependency on labeled data, some researchers have also applied unsupervised learning approaches, wherein variants of topic models and neural network-based models have been built for this task. These unsupervised methods focus on co-occurrences of words and ignore the contextual meaning of the words in the given sentence. Thus, such models perform reasonably well in detecting the explicitly expressed aspect category but often fail in identifying the implicitly expressed aspect category in the sentence. This paper focuses on the contextual meaning of the word. It adopts a document clustering approach requiring minimal user guidance, i.e., only a small set of seed words for each aspect category to detect efficiently implicit and explicit aspect categories. A novel BERT-based Attentive Representation Learning with Adversarial Training (BARLAT) model is presented in this paper, which utilizes domain-based contextual word embedding (BERT) for generating the sentence representation and uses these representations for clustering the sentences through attentive representation learning. Further, the model parameters are generalized better by performing adversarial training, which adds perturbations to the cluster representations. BARLAT is the first nearly unsupervised method that uses the contextual meaning of the words for learning the aspect categories through an adversarial attentive learning approach. The performance of BARLAT is compared with various state-of-the-art models using F1-score on Laptop and Restaurant datasets. The experimental results show that BARLAT outperforms the best existing model by a margin of 1.1% and 2.3% on Restaurant and Laptop datasets, respectively.

**Keywords** Aspect category detection · BERT · Deep neural network · Attention mechanism · Adversarial network · Unsupervised learning

✉ Avinash Kumar
p20150507@hyderabad.bits-pilani.ac.in

[1] Birla Institute of Technology and Science, Pilani - Hyderabad, Hyderabad 500078, India

[2] APP Centre for Artificial Intelligence Research, BITS Pilani Hyderabad Campus, Hyderabad, India

## 1 Introduction

Social media is one of the biggest platforms for people to express their opinion. Online reviews play a key role in influencing each individual's decision-making process. Aspect Based Sentiment Analysis (ABSA) [1–3] analyzes people's opinions and sentiments about a product or an event in a fine-grained manner. Opinion without knowing the target or topic provides limited insight into the user's viewpoint [4]. Aspect category detection is a key ABSA task that helps to identify the general topic or aspect category (i.e., characteristics of the product or service) discussed in the opinionated text. For example, in the given sentence *Everything from the pasta to the fish and even the chicken was very tasty.*, opinion is expressed about fine-grained aspect-terms like *pasta*, *chicken* and *fish*. Thus, *Food* should be identified as the aspect category for this review sentence. Identifying the aspect category in a review sentence helps to perform aspect-dependent sentiment analysis.

Previous work for aspect category detection can primarily be categorized into two approaches: supervised and unsupervised. Supervised methods use pre-defined aspect categories as labels and consider this task as a multi-label classification problem. Unsupervised methods are used to avoid dependency on labeled data. Among previous unsupervised approaches, Latent Dirichlet Allocation (LDA) [5] based topic modeling has been widely adopted for uncovering the hidden aspect categories. Many variants of LDA [6–8] have also shown promising results for this task. In general, LDA models the corpus as a mixture of topics (aspect categories), where each topic is a probability distribution over words. Conventional LDA models assume that each word is generated independently. It captures the latent topics within the corpus by implicitly encoding the document-level word co-occurrence patterns [9]. In general, review texts are short documents, and words in such short documents play a less discriminative role than lengthy documents. Thus, LDA-based approaches give poor quality of aspects, wherein detected aspect categories often consist of unrelated or loosely related concepts. Recently, neural network topic models [10–12] have been used for aspect category detection tasks and have shown better performance than LDA based approaches. He et al. in [13] present a state-of-the-art neural network-based approach that explicitly encodes the tokens into word-embeddings based on word-occurrence statistics and uses dimension reduction to extract the essential aspect categories. Tulkens and van Cranenburgh et al. [14] introduce a Radial Basis Function (RBF) kernel-based single-head attention approach that uses a POS tagger and domain-specific word embeddings. Both these models [13, 14] utilize Word2Vec [15] word embeddings, which provides a general static semantic meaning of the word, and use the same to encode the global context of the sentence. However, these approaches ignore the positional information and contextual meaning of a word in the sentence. Thus, the encoded sentence carries minimal information about the review sentence and can degrade the performance of the aspect category detection task.

Usually, the names of the categories are not explicitly mentioned in the review sentence. Thus, aspect categories are inferred by understanding the overall meaning of the sentence. For example, in the sentence, *We had to wait very long to get our food*, the user is expressing an opinion about *service* of a restaurant in an implicit way. Thus, to detect aspect categories precisely, a neural network needs to understand the overall meaning of a sentence comprehensively and encode the same efficiently. In this paper, the aspect category detection task is modeled as a text clustering problem and argues the following points to effectively encode the sentence-level information.

1. The position of a word in a sentence plays a vital role. For example, changing the position of word *amazing* in the review sentence *Staff served me amazing food* forms a new

sentence *Amazing staff served me food* which conveys a different meaning compared to the original sentence. The former sentence expresses the opinion about aspect category *staff* while the later, talks about *food*. Thus, **inclusion of positional information of words** is important for encoding the sentence.

2. The same word in different sentences can have a different meaning. For example, in the review sentences, *They delivered us food really fast* and *This restaurant is known for its yummy fast food*, word *fast* is used in two different contexts. In the first sentence, it is referring to *speed of service* while in the second sentence, it is used for *food item*. Hence, **considering the contextual meaning** of the word can enhance the quality of encoded sentence representation.

In this paper, motivated by the above-analyzed points, a nearly unsupervised neural topic model is proposed, which needs only a small set of seed words for each aspect category along with a corpus of an unlabeled text. The proposed model uses Bidirectional Encoder Representations from Transformers (BERT) [16] to obtain the contextual meaning of the word and encode the sentence level information. The proposed model adopts an adversarial attentive representation learning methodology to detect the aspect category in the sentence.

The main contributions of this work can be summarized as follows:

- A novel architecture called BERT-based Attentive Representation Learning with Adversarial Training (BARLAT) is proposed for aspect category detection in a nearly unsupervised way. BARLAT uses BERT to generate meaningful sentence representation and combines it with clustering in a unified framework to learn the aspect categories. This framework uses a regularization method to integrate user guidance in the learning process. Further, the model parameters are generalized in a better way by performing adversarial training.
- The effectiveness of adversarial training is investigated in BARLAT for the Aspect category detection task.
- The results of various baseline models are reproduced on two real-world datasets and compared with BARLAT. It is shown that the proposed model outperforms all other models.

The rest of this paper is organized as follows, after discussing related work in Sect. 2, a detailed description of the proposed BARLAT model is presented in Sect. 3. In Sects. 4 and 5, the details of extensive experiments are discussed along with the analysis of results. Finally, this work is summarized in Sect. 6.

## 2 Related Work

The work in aspect extraction and aspect category detection can be mainly categorized into three types- rule-based methods, supervised methods, and unsupervised methods. Initially, the studies focused on rule-based approaches where aspect terms were extracted by mining frequent noun terms or noun phrases [1]. The main drawback of such models is that it depends on manually defined rules that work only on noun terms and noun phrases, thus limiting the aspect term extraction to nouns only. Also, these models were not able to categorize the extracted aspects into various aspect categories.

Supervised methods typically model aspect extraction task as a sequence labeling task where each token in a sentence is labeled as one of Begin ($B$), Inside ($I$), Outside ($O$). A sequence of terms is called an aspect if it starts with $B$ followed by $O$ or more $I$'s. The major

drawback of supervised methods is that they require a large amount of annotated data for training purposes and restrict the model's scalability across domains since the annotations might be different for different domain datasets.

Unsupervised methods remove the need for labeled data for the task. Topic models like LDA [5] and its variants [6, 7, 17, 18] extract aspect terms by implicitly finding word co-occurrence patterns in the corpus at a document level. The latent topics or aspect categories are found by grouping the frequently co-occurring words. But they suffer from data sparsity problems since the review sentences are too small. This problem is addressed by BTM [19] which explicitly captures the word co-occurrence patterns at a global level rather than at the document level. However, this model fails to consider the semantic meaning of words while learning the topics or aspect categories.

More recently, neural method-based approaches and attention mechanisms have been used for aspect extraction and aspect category detection. He et al. [13] proposed the ABAE model where sentences are encoded using word embeddings after applying an attention mechanism. Aspect categories are found using dimension reduction of the encoded sentence in a training process similar to autoencoders. Jure et al. [20] propose an objective function to avoid redundancy between components of embedding vectors by measuring the cross-correlation matrix between the outputs of two identical networks fed with distorted versions of a sample. Liao et al. [12] proposed a neural network-based model to extract aspect terms by coupling the global and local context of words in a sentence. This model uses an LSTM layer to encode the word sequences, which helps capture the positional information of words but is not quite effective in capturing contextual information. Tulkens et al. [14] introduced a single head attention-based CAT model that uses Radial Basis Function (RBF) kernel for detecting the aspect category.

The above-mentioned neural network-based methods used the static meaning and ignore the positional information and contextual meaning of a word in the sentence. Hence, the encoded sentence carries very limited information about the review sentence and can degrade the model's performance. These models also ignore the generalization of model parameters.

The aspect category detection task can also be formulated as a text clustering task wherein sentences belonging to the same aspect category are classified in the same cluster. Recently, many neural-based clustering methods have been developed to cluster short texts [21, 22]. Zhang et al. [23] proposed a neural network-based method that uses cluster-level attention and adversarial training for text clustering. However, they ignore the importance of contextual meaning and positional information of the word while generating the sentence representation for sentence clustering. Table 1 summarizes the previous works.

## 3 Proposed Scheme

This section describes the architecture of the proposed model, BERT Attentive Representation Learning with Adversarial Training (BARLAT). As illustrated in Fig. 1, BARLAT contains sentence encoding using BERT word embedding, cluster-level attention, sentence reconstruction, adversarial perturbation, and objective functions to be optimized. BARLAT adopts an **attentive representation learning** methodology to generate sentence representation and learn the clustering of review sentences in a unified manner. Cluster-level attention is applied in the proposed model to capture the correlation between a given sentence and each cluster. Subsequently, the resultant attention weights are used to determine cluster assignments. The parameters of the proposed model are trained in an unsupervised way by reconstructing the

**Table 1** Summary of related works

| Approaches | Summary |
|---|---|
| Rule-based methods [1, 2, 4] | These methods focus on lexicons and dependency relations and utilize manually defined rules to identify patterns and extract aspects. The drawback of these methods is that it requires domain-specific knowledge or human expertise |
| Statistical machine learning based methods [5–7, 9, 17–19] | Approaches mentioned in these works utilize information related to word co-occurrences at the document level to identify aspects. However, these methods suffer from data sparsity problems and often gives a poor quality of aspects |
| Neural network based methods [12, 13, 21–26] | These methods have shown strong performance in extracting coherent aspects and clustering the sentences. The main drawback with these methods is that it ignores the contextual meaning of a word and lacks in generalizing the model parameters |

sentence-level representation through the weighted combination of cluster embeddings. In the proposed model, a regularization is applied to the cluster embeddings to instill the user guidance so that the learned cluster embeddings remain close to the seed words in the embedding space. The training process for aspect category or cluster embeddings is analogous to autoencoder. The performance of the proposed model is improved further by adding adversarial examples, which are getting generated by applying small perturbations to the cluster embeddings. The inclusion of such adversarial examples during training mislead a neural network and force it to behave incorrectly [27]. Thus, **adversarial training** plays a role of adaptive regularization and helps the proposed model to generalize better [28]. The ultimate goal of BARLAT is to learn a set of cluster embeddings, where a group of seed words guides the learning process.

### 3.1 Sentence Encoding

We use Bidirectional Encoder Representations from Transformers (BERT) [16] to encode the input sentence $I$ that consists of $m$ tokens $w_i$, where $i \in [1, \mathrm{m}]$. BERT is trained using two objectives: "Masked Language Model" (MLM) and "Next Sentence Prediction" (NSP) pre-training objective. The MLM, masks some of the tokens from the input before feeding them into BERT. The objective is to predict the original value of the masked words based on the context provided by the other non-masked words in the sequence. In the NSP, BERT receives pairs of sentences as input and predicts whether the second sentence in the pair semantically follows the first sentence from the pair. MLM and NSP allow BERT to capture token-level and sentence-level information, respectively. This helps BERT to generate word embeddings which are very rich in representing contextual information. BERT calculates the input representation for each token by summing over token, position, and segment embeddings. Token Embeddings are generated using WordPiece Embeddings [29], which provide vocabulary IDs to BERT. Positional Embeddings indicate the position of a word in each sentence. Segment embeddings are used to distinguish among sentences. For a task like question answering, where more than two sentences are given as input, segment embeddings provide the same label to all the sentence tokens. The $BERT_{BASE}$ model structure consists of a 12-layer bidirectional Transformer encoder [30].

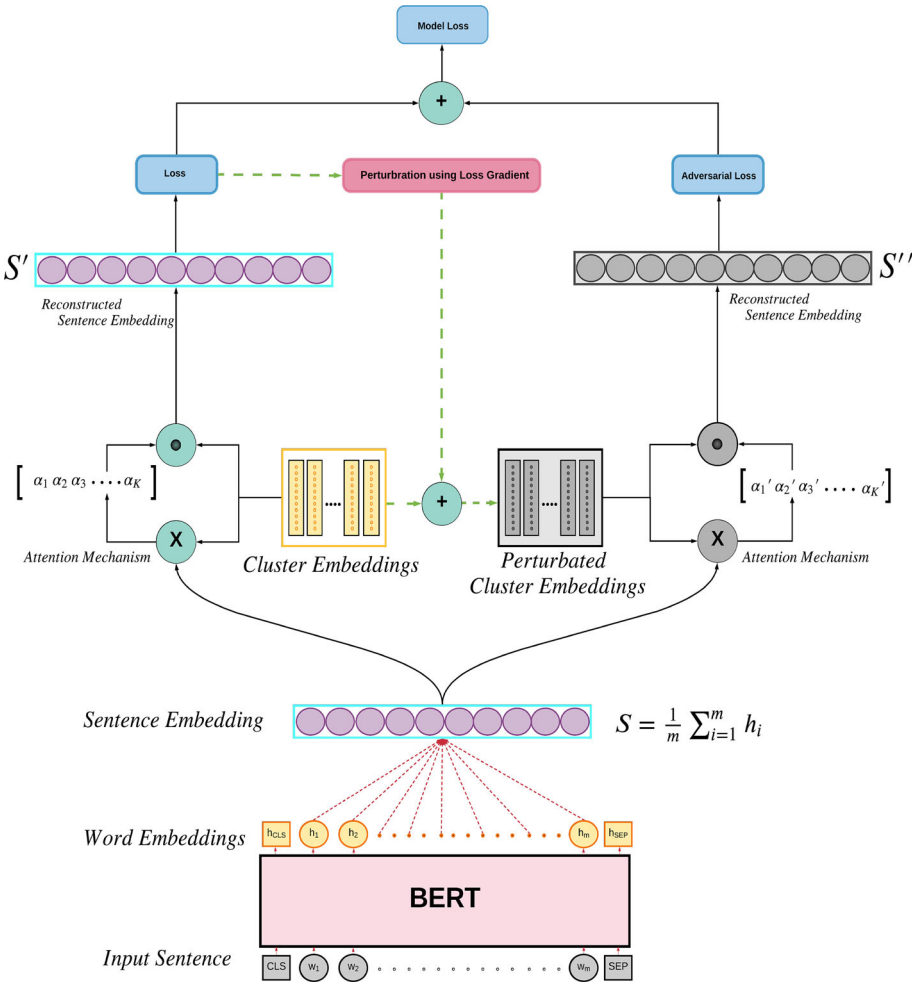



**Fig. 1** Architecture of the proposed model for aspect-category detection

In our proposed method we are using $BERT_{BASE}$ only to generate word embedding without doing fine-tuning. The output of the last layer could be too close to BERT's own target function (i.e. next sentence prediction and masked language model). Thus, the proposed model uses the output of the second-to-last hidden layer (i.e., eleventh encoder) as the embeddings [31] of all of the tokens in the sentence. For a sentence $I$, the input sequence $X$ is constructed as:

$$X = ([CLS], w_1, w_2, \ldots w_m, [SEP]), \tag{1}$$

Here, [CLS] and [SEP] are special tokens that indicate the beginning and end of the sequence, respectively. The constructed sequence $X$ has length $m + 2$. This sequence is passed to the BERT tokenizer which, generates tokens corresponding to each word. Subsequently, such tokens are passed to the encoder layer of BERT to obtain hidden representation $H \in \mathbb{R}^{(m+2)\times d}$

for all the items of $X$.

$$H = BERT(X), \tag{2}$$

Here, $d$ is the size of the hidden dimension of BERT. The proposed model takes hidden representation $h_i \in H$ for each token (except [CLS] and [SEP] tokens) of the input sequence $X$, and perform the mean-pooling technique to construct the contextual sentence representation $S$:

$$S = \frac{1}{m}\sum_{i=1}^{m} h_i, \tag{3}$$

### 3.2 Cluster Attention

It is assumed that $K$ aspect categories exist in the given corpus, where each aspect category represents a cluster and, a cluster embedding matrix $C \in \mathbb{R}^{d \times K}$ is defined.

$$C = [c_1, c_2, \ldots c_K], \tag{4}$$

Sentence representation $S$ contains useful global semantic information for a review sentence, and to identify the most relevant cluster for $S$, cluster-level attention is applied. For each cluster $c_i$, a positive weight $\alpha_i$ is computed, which can be interpreted as the probability that $c_i$ is a suitable cluster for the global context of the sentence. The probability $\alpha_i$ is computed by an attention mechanism, which uses the embedding of cluster $c_i$ and the global context of the sentence $S$.

$$\alpha_i = \frac{\exp(c_i^T S)}{\sum_{j=1}^{K} \exp(c_j^T S)}, \tag{5}$$

The above Eq. (5) first calculates the semantic similarity of a sentence with each of the $K$ aspect categories through a dot product. Subsequently, using the values of semantic similarities, it computes the probability score $\alpha_i$ that shows the probability of sentence $S$ belonging to an aspect category $c_i$. The proposed model uses these probability scores during prediction and classifies the given sentence to an aspect category that contains the highest probability score.

### 3.3 Sentence Reconstruction with Cluster Embedding

Sentence clustering is an unsupervised learning problem. The proposed model leverages cluster embeddings to reconstruct the sentence. The sentence is reconstructed as a linear combination of the sentence-dependent cluster embedding.

$$S' = \sum_{i=1}^{K} \alpha_i c_i, \tag{6}$$

Using the attention score $\alpha_i$ of the cluster $c_i$, the proposed model rebuilds an expressive representation $S'$ of the sentence $S$. The cluster-level attention mechanism gives more attention scores to those clusters that are more semantically similar to the sentence representation $S$. In other words, a review sentence which is talking about *food* of a Restaurant will have higher semantic similarity with the *food* aspect category compared to other aspects categories like *price* or *ambiance*. Hence, each aspect category or cluster will have different attention scores,

and the cluster with the higher attention score will contribute more to the reconstructed sentence $S'$ in Eq. (6). During the training process, as the reconstructed sentence $S'$ is driven to be closer to the original sentence representation $S$, the cluster level attention may ideally start favoring one single cluster.

### 3.4 Objective Function

BARLAT is trained to minimize the reconstruction error and uses max-margin loss as used in previous works [32–34]. For each input sentence, pseudo negative samples $N_a$, where $a \in [1, q]$ are generated through random sampling of training data. Sentence encoding of these negative samples is done by using the mean-pooling technique of its word embedding. The proposed model aims to make the reconstructed embedding $S'$ similar to the original sentence embedding $S$ and entirely dissimilar to those negative sample set $N_a$. To achieve this objective, $L$ is formulated:

$$L(i; \mathbf{C}) = \sum_{a=1}^{q} \max(0, m - S'S + S'N_a), \tag{7}$$

The above Eq. 7 calculates the reconstruction loss of $i$-th sentence in the corpus $D$, which ensures a larger semantic similarity between $S'$ and $S$ and larger dissimilarity with negative samples. We experiment with different values of margin, $m$ and set it to 1 in the above equation 7 (discussed in Sect. 5.2).

*Seed regularization* A small set of user-guided seed words for each aspect category are used to apply a regularization on the cluster embeddings $C$. All such sentences are taken from the training dataset, where user-guided seed words are present and corresponding to each sentence, BERT embedding of the seed word is obtained. A matrix $R \in \mathbb{R}^{d \times K}$ is created of the same size as parameter $C$. The $j$-th row of $R$ is assigned with the average embedding of seed words in the corresponding aspect category. The proposed model applies a regularization on the cluster embeddings $C$ using the matrix $R$. The regularization aims to make each row of the cluster embeddings $C$ semantically closer to the corresponding row of the matrix $R$ and penalize learned $j$-th aspect embedding (represented by the $j$-th row of $C$) when it significantly differs from the average embedding of seed words. Accordingly, the regularization term for $i$-th sentence in the corpus $D$ is defined as:

$$Q(i; \mathbf{C}) = \sum_{j=1}^{K} [1 - R_j C_j], \tag{8}$$

*Redundancy regularization* The aspect embedding $C$ is prevented from having almost identical rows. A penalty is applied on the redundancy of learned aspect embedding $C$. The regularization term is:

$$U(i; \mathbf{C}) = ||C_n C_n^T - I||, \tag{9}$$

The above Eq. 9 computes the L2 norm of the redundancy regularization loss for $i$-th sentence in the corpus $D$, where $C_n$ is $C$ with each row normalized to a unit length vector, and $I$ is the identity matrix. The final objective function with regularizations is:

$$\lambda(i; \mathbf{C}) = L(i; \mathbf{C}) + \beta_1 Q(i; \mathbf{C}) + \beta_2 U(i; \mathbf{C}), \tag{10}$$

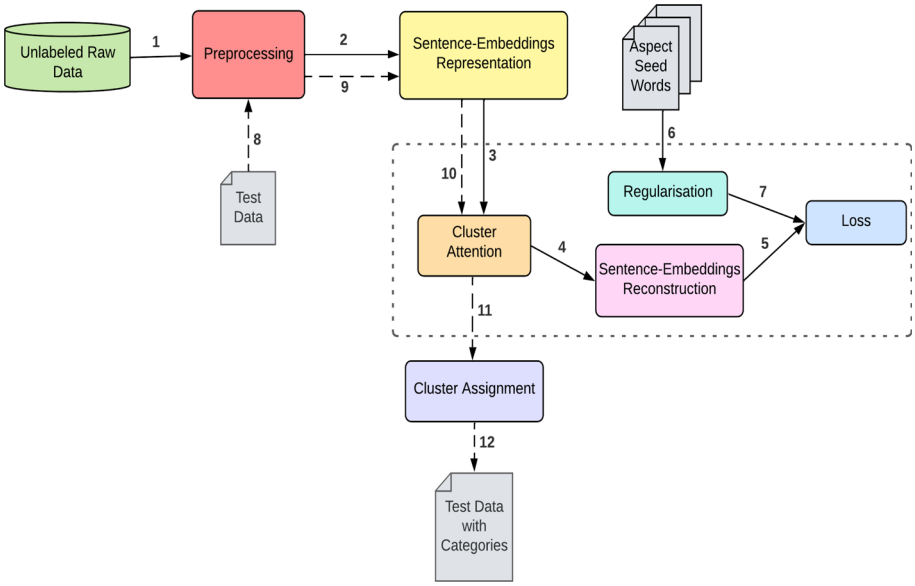



**Fig. 2** Flowchart of the proposed model

Here, $\beta_1$ and $\beta_2$ are hyperparameters. Considering the total number of sentences $D$ in the training set, the optimization function is defined as follows:

$$E_1(\mathbf{C}) = \sum_{i=1}^{D} \lambda(i; \mathbf{C}), \tag{11}$$

Here, $C$ represents the trainable parameters of BARLAT.

### 3.5 Adversarial Training

Adversarial training confuses neural network-based models to make erroneous predictions. In this step, the parameters of the models are generalized in a better way. In general, adversarial training provides a new objective function based on adversarial perturbations to complement the original optimization procedure. In the proposed network, perturbations are created using the gradient of the original loss function. Following the work done by Miyato et al. in [35], adversarial perturbations are added to the cluster embeddings. The intuition behind not adding adversarial perturbations to the word embedding is that the number of words is much larger than that of clusters, so adding perturbations to words needs more parameters instead of clusters. However, during empirical validation, it is found that adding perturbations to words embedding of BERT degrades the performance of BARLAT compared to adding perturbations to cluster embedding as shown in Table 7. Using the cluster-level adversarial perturbations $\nabla_c \in \mathbb{R}^{d \times K}$, the following objective function is used in BARLAT:

$$E_2(\mathbf{C}) = \sum_{i=1}^{D} \lambda(i; \mathbf{C} + \nabla_c), \tag{12}$$

Goal of above objective function $E_2(\mathbf{C})$ is to achieve the worst perturbations $\nabla_c$ while optimizing the model parameters $\mathbf{C}$. By combining the two objective functions, $E_1$ and $E_2$, the final objective function of the proposed model is defined as follows:

$$E(\mathbf{C}, \nabla_c) = E_1(\mathbf{C}) + \beta_3 E_2(\mathbf{C} + \nabla_c), \tag{13}$$

In the above target objective function $E$, $E_2$ acts as a regularizer and compliments $E_1$. Strength of $E_2$ is controlled by $\beta_3$. The learning of adversarial perturbation $\nabla_c$ can be approximated by linearizing the methodology proposed by Goodfellow et al. in [28]. Therefore, by making use of $L2$ norm, each column of $\nabla_c^k \in \nabla_c$, where $k \in [1, K]$, are added to the cluster embeddings to create new adversarial aspect categories in the embedding space.

$$\nabla_c^k = \epsilon \frac{y_c^k}{||y_c^k||}, \tag{14}$$

where,

$$y_c^k = \frac{\partial E_1}{\partial C_k}, \tag{15}$$

In the Eq. (14), $\epsilon$ is the size of the perturbations. An ablation study is carried out on many values for epsilon to find the values which outperform the original results. These results are discussed in Sect. 5.2. Figure 2 shows the set of operations during pre-processing, training, and testing phases. Algorithm (1) summarizes the overall working of BARLAT.

**Algorithm 1:** BARLAT

**Input**: Corpus $\mathcal{D}$, number of aspects $K$, $A_i$: small set of seed words for the $i^{th}$ aspect category $i \in [1, K]$, $\beta_1, \beta_2, \beta_3$ hyperparameters
**Output**: $C$: trained cluster embedding matrix

```
/* Initialize aspect prior R                                    */
for i ∈ [1, K] do
    Encoding and averaging seed words in A_i using BERT to obtain i^th aspect prior representation g_i ∈ ℝ^d
    R_i ← g_i
end
Randomly initialize cluster embedding matrix C
for iter = 1 to max_iter do
    Sample a mini-batch D_batch ⊂ D
    Randomly sample negative examples T_batch ⊂ D
    for a ∈ T_batch do
        Compute N_a using Eq. 2 & 3
    end
    E_1(C) ← 0
    E_2(C) ← 0
    for X ∈ D_batch do
        Compute S using Eq. 3
        Compute S' using Eq. 6
        Compute loss using Eq. 10 and update E_1(C)
        Learn adversarial perturbations using Eq. 14 & 15
        Compute adversarial loss and update E_2(C)
    end
    Compute total loss using Eq. 13
    Compute gradients and update cluster embedding matrix C
end
```

**Table 2** Dataset statistics

| Dataset | #Training reviews | #Test reviews |
|---|---|---|
| Restaurant | 52,574 | 3400 |
| Laptop | 14,683 | 307 |

## 4 Experiments

### 4.1 Datasets

The proposed model is domain-independent and can be applied to detect the aspect categories in various domains of review comments. To provide conclusive evidence about the same, two datasets of different domains are chosen for the experiment. The statistics of both the datasets are shown in Table 2.

*Restaurant* This dataset is from Citysearch New York[1] which is widely used by previous works [13, 36]. It contains more than 50,000 restaurant reviews in training and an annotated 3,400 reviews in the test dataset. This work follows the experimental settings of previous work [6, 13, 36], and uses the single-label sentences for evaluation to avoid ambiguity. Additionally, in order to compare with other systems, this work adopts restricted experimental settings to three labels: *Food*, *Service*, and *Ambiance*.

*Laptop* This dataset contains 14,683 unlabeled Amazon reviews under the laptop category collected in [37] for training. The test dataset comprises of 307 reviews from the Laptop domain of the benchmark dataset of SemEval-2016 and SemEval-2015. The test dataset has eight labels: *Support*, *OS*,*Display*, *Battery*, *Company*, *Mouse*, *Software* and *Keyboard*.

### 4.2 Baseline Methods

To validate the performance of BARLAT, it is compared against several baselines. These baselines are divided into two groups. The first group of the baselines are built using a traditional statistical machine learning approach, whereas the second group of baselines are built using neural network-based architecture. The results of some baselines are reproduced using the respective official codebase. Below are the details of the **first group of baseline models**:

- *LDA* [6] This method uses a standard implementation of LDA and treats each sentence as a separate document to infer the aspect categories.
- *BTM* [19] This is a biterm topic model that is specially designed for short texts such as texts from social media and review sites.
- *Seeded BTM* [38] Similar to BTM [19], this model also tackles the data sparsity problem during aspect extraction from short documents by modeling the review corpus as a collection of biterms, which is a pair of words that have co-occurred in some review sentence. The model then tries to learn the corpus-aspect distribution and aspect-word distribution by introducing seed sets, thus, encoding domain-specific knowledge to guide the model.
- *CosSim* [39] In this model, the aspect category representations created by taking the average of the embeddings of seed words. Likewise, sentence representation is generated

[1] http://www.cs.cmu.edu/~mehrbod/RR/.

**Table 3** Seed words for each aspect category

| Dataset | Aspect category | Seed words |
|---|---|---|
| Restaurant | Food | Delicious, bland, drinks, flavourful, spicy |
| | Staff | Manager, waitress, rude, forgetful, server, quick |
| | Ambience | Atmosphere, seating, surroundings, environment, decoration |
| Laptop | Support | Service, warranty, issues, customer, coverage |
| | OS | Windows, mac, linux, android, ios |
| | Display | Monitor, resolution, desktop, screen, led |
| | Battery | Power, charge, life |
| | Company | Product, apple, microsoft, samsung, dell |
| | Mouse | Wireless, pad, cursor |
| | Software | Application, program, spreadsheet, presentation |
| | Keyboard | Mechanical, keys, strokes, input |

by taking the average of the embeddings of words. To classify the sentence, cosine similarity is computed between the representation of the test sentence and the aspect category representations. This model is built using Word2Vec.

- *BERT K-means* [16] Generates sentence representation using BERT word embedding and then applies the $K$-means algorithm.
- *SERBM* [36] is a Restricted Boltzmann Machine (RBM) based model that learns topic distributions and assigns individual words to these distributions.
- *W2VLDA* [24] is a topic modeling approach that biases word-aspect associations by computing the similarity from a word to a set of aspect terms.
- *CAT* [14] This is a very recent Radial Basis Function (RBF) kernel-based single-head attention approach that requires a POS tagger and in-domain word embeddings for aspect category detection.
- *SUAEx* [40] This is also a recent model, which does not use a neural network architecture and relies on the similarity of word-embeddings and on reference words to emulate the attention mechanism used by attention neural networks. Below are the details of **second group of baseline models**, which are built using neural network-based architecture:
- *ABAE* [13] This is a neural-based approach to extract aspect categories from the corpus. It uses Word2Vec word embeddings to encode the sentence after applying an attention mechanism. Aspect categories are extracted using dimension reduction of the encoded sentences in a training process similar to that of autoencoders.
- *ABAE-init* [25] This is built by replacing each aspect embedding vector in ABAE with the corresponding centroid of seed word embeddings. During the training, aspect embedding vectors remains fixed.

– *MATE* [25] This uses the weighted average of seed word embeddings to initialize aspect embeddings to build another version of ABAE.
– *AE-CSA* [41] This is similar to ABAE. In addition to word vectors and aspect vectors, this model also considers sense and sememe vectors in computing the attention distribution.
– *CMAM* [26] This is a neural network-based model that applies the convolutional multi-attention mechanism for aspect category detection.

### 4.3 Variants of the Proposed Model

To verify the effectiveness of various components of BARLAT, the following variants of the proposed model are built. These models are also used in ablation studies.

– *BARLAT* This is the proposed model, which uses domain knowledge BERT word embedding and optimizes the objective function as mentioned in Eq. (13).
– *BARLAT (word)* This model is very similar to BARLAT, except that adversarial perturbation is added to BERT word embeddings instead of cluster embeddings.
– *BARLAT w/o adv* Adversarial training is removed from the proposed model. It optimizes the objective function $E_1$ represented in Eq. (11) as optimization target.
– *BARLAT w/o adv & redundancy reg.* In addition to the above model, redundancy regularisation (as described in Eq. 9) is removed from the objective function during training.
– *BARLAT w/o adv & redundancy reg. & seed word reg.* In addition to the above model, seed word regurlarisation (as described in Eq. 8) is also removed from the objective function during training.
– *BARLAT w/o adv & redundancy reg. & seed word reg. & DK-BERT* This model uses $BERT_{BASE}$ instead of domaing knowledge BERT in the above model.

### 4.4 Settings

The datasets are preprocessed by removing punctuation symbols and stop words. Following the work of Xu et al. in [42], BERT with Restaurant and Laptop domain knowledge (DK-BERT) is used to generate word embeddings for each dataset. Word embedding size $d$ is 768. The regularization matrix $R$ is initialized with the seed words given in Table 3 for each aspect category. For optimization, Adam optimizer [43] is used with learning rate 0.001. The number of negative samples per input sample $q$ is set to 20. Contribution of seed and redundancy regularization in the Eq. 10 is controlled by setting value of hyperparameters $\beta_1$ and $\beta_2$ as 10 and 0.1 respectively. The value of $\beta_3$ is set as 1 to give equal weight to standard loss and adversarial loss and calculate the overall loss of the model. The proposed model is trained for 35 epochs and uses a batch size of 32. The results reported for all models are the average over five runs.

### 4.5 Evaluation Metrics

In this work, $precision$, $recall$, and $F1\text{-}score$ are used to understand how well predictions match with the true labels for each aspect category. To understand the overall performance of a model across all the aspect categories, macro-averaged precision, macro-averaged recall, and macro-averaged F1-score are calculated as the evaluation measures.

**Table 4** Aspect category identification results on the Restaurant domain

| Aspect | Method | Precision | Recall | F1-score |
|---|---|---|---|---|
| Food | LDA | 89.8 | 64.8 | 75.3 |
| | BTM | 93.3 | 74.5 | 81.6 |
| | Seeded BTM | 81.5 | 82.4 | 81.9 |
| | CosSim | 77.8 | 79.3 | 78.5 |
| | BERT k-means | 85.0 | 74.2 | 79.2 |
| | SERBM | 89.1 | 85.4 | 87.2 |
| | W2VLDA | **96.0** | 69.2 | 81.1 |
| | CAT | 91.8 | 92.4 | 92.1 |
| | SUAEx | 91.7 | 90.0 | 90.8 |
| | ABAE | 95.3 | 74.1 | 82.8 |
| | ABAE-init | 95.8 | 77.3 | 85.6 |
| | MATE | 94.9 | 80.1 | 86.9 |
| | AE-CSA | 90.3 | 92.6 | 91.4 |
| | CMAM | 88.7 | **94.5** | 91.5 |
| | BARLAT | 94.1 | 92.4 | **93.2** |
| Service | LDA | 80.4 | 58.5 | 67.7 |
| | BTM | 83.0 | 58.1 | 67.9 |
| | Seeded BTM | 83.4 | 58.5 | 68.8 |
| | CosSim | 71.1 | 69.2 | 70.1 |
| | BERT k-means | 82.7 | 63.2 | 71.6 |
| | SERBM | 81.9 | 58.2 | 68.0 |
| | W2VLDA | 61.1 | 85.9 | 71.0 |
| | CAT | 82.4 | **75.6** | 78.8 |
| | SUAEx | 66.0 | 87.2 | 75.2 |
| | ABAE | 80.2 | 72.8 | 75.7 |
| | ABAE-init | 80.5 | 75.0 | 77.6 |
| | MATE | 81.3 | 74.2 | 77.5 |
| | AE-CSA | 80.4 | 75.5 | 77.3 |
| | CMAM | 80.5 | 67.6 | 73.5 |
| | BARLAT | **91.7** | 75.3 | **82.7** |
| Ambience | LDA | 60.3 | 67.7 | 63.8 |
| | BTM | 81.3 | 59.9 | 68.5 |
| | BTM-Seed | 82.4 | 60.7 | 69.9 |
| | CosSim | 80.1 | 63.2 | 70.6 |
| | BERT k-means | 78.9 | 66.9 | 72.4 |
| | SERBM | 80.5 | 59.2 | 68.2 |
| | W2VLDA | 55.1 | 75.2 | 64.1 |
| | CAT | 76.6 | **80.1** | 76.6 |
| | SUAEx | 88.4 | 54.6 | 67.5 |
| | ABAE | 81.5 | 69.8 | 74.0 |
| | ABAE-init | 82.0 | 73.1 | 77.3 |
| | MATE | 81.7 | 74.0 | **77.6** |

**Table 4** continued

| Aspect | Method | Precision | Recall | F1-score |
|---|---|---|---|---|
| | AE-CSA | **91.4** | 77.9 | 77.0 |
| | CMAM | 81.3 | 59.9 | 68.5 |
| | BARLAT | 78.4 | 70.2 | 74.1 |

Bold shows the numbers of best performing models against each of the metrics

**Table 5** Aspect category identification results for the Laptop domain

| Aspect | Method | Precision | Recall | F1-score |
|---|---|---|---|---|
| Support | LDA | 62.3 | 46.7 | 46.7 |
| | BTM | 72.1 | 46.9 | 56.8 |
| | BTM-Seed | **72.6** | 47.1 | 57.1 |
| | CosSim | 52.9 | 39.1 | 45.0 |
| | BERT k-means | 45.9 | 63.1 | 53.1 |
| | SERBM | – | – | – |
| | W2VLDA | – | – | – |
| | CAT | 57.3 | 67.6 | 62.1 |
| | SUAEx | – | – | – |
| | ABAE | 61.6 | 48.2 | 54.1 |
| | ABAE-init | 62.1 | 50.3 | 55.4 |
| | MATE | 61.7 | 52.6 | **56.2** |
| | AE-CSA | – | – | – |
| | CMAM | 59.3 | 68.9 | 63.7 |
| | BARLAT | 60.4 | **70.5** | **65.1** |
| OS | LDA | 57.3 | 50.7 | 53.3 |
| | BTM | 65.4 | 51.6 | 57.8 |
| | BTM-Seed | 66.6 | 52.3 | 58.6 |
| | CosSim | 58.3 | 46.6 | 51.8 |
| | BERT k-means | 69.8 | 54.9 | 61.4 |
| | SERBM | – | – | – |
| | W2VLDA | – | – | – |
| | CAT | 67.5 | 59.5 | 63.2 |
| | SUAEx | – | – | – |
| | ABAE | 74.1 | 61.2 | 65.9 |
| | ABAE-init | 74.8 | 60.3 | 65.7 |
| | MATE | 75.7 | 60.6 | 66.4 |
| | AE-CSA | – | – | – |
| | CMAM | 73.4 | **61.9** | 67.2 |
| | BARLAT | **76.3** | 61.1 | **67.8** |
| Display | LDA | 40.3 | 51.6 | 45.3 |
| | BTM | 47.4 | 78.6 | 59.1 |

**Table 5** continued

| Aspect | Method | Precision | Recall | F1-score |
|---|---|---|---|---|
| | BTM-Seed | 50.0 | **81.3** | 61.9 |
| | CosSim | 54.7 | 61.0 | 57.7 |
| | BERT k-means | 71.3 | 53.0 | 60.8 |
| | SERBM | – | – | – |
| | W2VLDA | – | – | – |
| | CAT | 81.4 | 78.6 | 80.0 |
| | SUAEx | – | – | – |
| | ABAE | 57.4 | 59.3 | 58.3 |
| | ABAE-init | 60.2 | 60.8 | 60.6 |
| | MATE | 58.7 | 60.6 | 59.6 |
| | AE-CSA | – | – | – |
| | CMAM | **82.3** | 80.1 | **81.2** |
| | BARLAT | 64.1 | 80.4 | 71.4 |
| Battery | LDA | 42.7 | 66.6 | 50.0 |
| | BTM | 61.6 | 78.0 | 68.8 |
| | BTM-Seed | 64.1 | 80.4 | 71.3 |
| | CosSim | 67.8 | 71.3 | 69.5 |
| | BERT k-means | 79.5 | 68.6 | 73.7 |
| | SERBM | – | – | – |
| | W2VLDA | – | – | – |
| | CAT | 80.1 | 82.3 | 81.2 |
| | SUAEx | – | – | – |
| | ABAE | 83.3 | 74.9 | 78.9 |
| | ABAE-init | 85.2 | 76.2 | 80.6 |
| | MATE | 83.9 | 77.5 | 81.4 |
| | AE-CSA | – | – | – |
| | CMAM | 79.4 | **87.1** | **83.1** |
| | BARLAT | **84.2** | 76.3 | 79.9 |
| Company | LDA | 45.8 | 53.5 | 49.4 |
| | BTM | 61.1 | 50.4 | 55.2 |
| | BTM-Seed | 63.9 | 51.2 | 56.8 |
| | CosSim | 56.2 | 47.6 | 51.5 |
| | BERT k-means | 64.5 | 57.6 | 60.9 |
| | SERBM | – | – | – |
| | W2VLDA | – | – | – |
| | CAT | 72.6 | 63.3 | 67.6 |
| | SUAEx | – | – | – |
| | ABAE | 72.8 | 54.9 | 62.6 |
| | ABAE-init | 73.7 | **64.1** | 68.6 |

**Table 5** continued

| Aspect | Method | Precision | Recall | F1-score |
|---|---|---|---|---|
| | MATE | 76.9 | 60.5 | 67.8 |
| | AE-CSA | – | – | – |
| | CMAM | 80.2 | 60.1 | 68.8 |
| | BARLAT | **87.1** | 61.2 | **71.9** |
| Mouse | LDA | 60.8 | 43.5 | 50.7 |
| | BTM | 75.1 | 45.3 | 56.5 |
| | BTM-Seed | 75.7 | 48.4 | 59.1 |
| | CosSim | 65.7 | 54.5 | 59.6 |
| | BERT k-means | 65.1 | 51.2 | 57.4 |
| | SERBM | – | – | – |
| | W2VLDA | – | – | – |
| | CAT | 71.4 | 57.1 | 63.4 |
| | SUAEx | – | – | – |
| | ABAE | 69.2 | 53.7 | 60.5 |
| | ABAE-init | 74.7 | 63.2 | 68.5 |
| | MATE | 73.6 | 60.3 | 66.4 |
| | AE-CSA | – | – | – |
| | CMAM | 82.2 | 66.8 | 73.7 |
| | BARLAT | **93.2** | **75.1** | **83.2** |
| Software | LDA | 54.7 | 33.9 | 41.9 |
| | BTM | 60.8 | 32.3 | 42.1 |
| | BTM-Seed | 57.4 | 38.7 | 46.2 |
| | CosSim | 54.3 | 58.4 | 56.3 |
| | BERT k-means | 67.3 | 51.5 | 58.3 |
| | SERBM | – | – | – |
| | W2VLDA | – | – | – |
| | CAT | 53.5 | 75.0 | 62.5 |
| | SUAEx | – | – | – |
| | ABAE | 54.2 | 70.1 | 61.1 |
| | ABAE-init | 53.7 | 74.2 | 62.3 |
| | MATE | 60.2 | **76.3** | 67.3 |
| | AE-CSA | – | – | – |
| | CMAM | 69.2 | 66.8 | 68.0 |
| | BARLAT | **74.3** | 70.2 | **72.3** |
| Keyboard | LDA | 54.7 | 52.9 | 53.8 |
| | BTM | 63.2 | 57.4 | 60.1 |
| | BTM-Seed | 64.3 | 58.3 | 61.2 |
| | CosSim | 68.0 | 53.1 | 59.6 |
| | BERT k-means | 60.7 | 49.2 | 54.3 |

**Table 5** continued

| Aspect | Method | Precision | Recall | F1-score |
|---|---|---|---|---|
| | SERBM | – | – | – |
| | W2VLDA | – | – | – |
| | CAT | 54.0 | 84.3 | 65.8 |
| | SUAEx | – | – | – |
| | ABAE | 57.1 | 70.2 | 62.9 |
| | ABAE-init | 60.0 | 73.9 | 66.2 |
| | MATE | 60.2 | 74.3 | 66.5 |
| | AE-CSA | – | – | – |
| | CMAM | 57.3 | **84.7** | 68.3 |
| | BARLAT | **60.7** | 78.6 | **68.5** |

Bold shows the numbers of best performing models against each of the metrics
The symbol '–' indicates the source code is not available in the respective papers. Hence, results are not reproduced

## 5 Results and Discussion

The performance of all the models for each aspect category on Restaurant and Laptop datasets are shown in Tables 4 and 5, respectively. The comparison of the $F1\text{-}score$ shows that BARLAT outperforms all other models on *Food* and *Service*, aspect categories in the Restaurant dataset. Likewise, its performance is better than all other models on *Support*, *OS*, *Company*, *Mouse*, *Software* and *Keyboard* in Laptop dataset. The performance of all the models across aspect categories is reported by calculating macro-averaged precision, macro-averaged recall and macro-averaged F1-score on both the datasets in Table 6. CosSim model classifies a given sentence by computing cosine similarity between representation of the test sentence and the aspect category representations. Its performance is lower than other models except for LDA. Usually, review sentences are short, which leads to data sparsity problems. BTM tackles the data sparsity problem by capturing the word co-occurrence patterns in the entire corpus. Seed-BTM uses seed sets to encode domain-specific knowledge to enhance the performance of BTM. The performance of Seed-BTM is better than LDA, BTM, and CosSim. However, its performance is comparable to BERT-K-means on both the datasets. SERBM is a restricted boltzmann machine based model and W2VLDA leverages keywords for each aspect for doing aspect classification. Performance of both these models are comparable but inferior to SUAEx and CAT, which are very recent attention-based methods and do not use neural network architectures.

CAT performs better than SUAEx and other attention-based baselines like ABAE, ABAE-init, MATE, and AE-CSA on both datasets. CAT uses a domain-specific general static meaning of the word for generating the sentence representations. CMAM is the latest neural network-based model that uses a convolutional multi-attention mechanism. CAT and CMAM perform better than other baselines on both the datasets, respectively. Macro-averaged F1-score comparison between BARLAT with these two strong baselines shows that BARLAT outperforms CAT and CMAM by a margin of 1.1% and 2.3% on Restaurant and Laptop datasets, respectively. The better performance of BARLAT compare to other baselines might be attributed to the following reasons: (1) BARLAT utilizes BERT based word embedding, which uses the contextual meaning of the word to obtain sentence embedding; (2) BARLAT

**Table 6** The weighted macro averages of various methods on Restaurant & Laptop datasets

| Method | Restaurant | | | Laptop | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1-score | Precision | Recall | F1-score |
| LDA | 78.3 | 65.7 | 71.4 | 59.6 | 54.1 | 50.8 |
| BTM | 83.4 | 70.4 | 76.3 | 62.3 | 57.6 | 57.1 |
| BTM-Seed | 82.1 | 73.0 | 76.7 | 63.5 | 59.9 | 59.8 |
| CosSim | 65.6 | 55.7 | 59.3 | 59.4 | 54.0 | 56.3 |
| BERT k-means | 83.4 | 70.3 | 76.2 | 66.1 | 56.3 | 60.4 |
| SERBM | 83.8 | 67.6 | 74.9 | – | – | – |
| W2VLDA | 72.7 | 75.1 | 73.9 | – | – | – |
| CAT | 86.5 | **86.4** | 86.4 | 69.5 | 71.4 | 69.7 |
| SUAEx | 83.2 | 78.1 | 80.6 | – | – | – |
| ABAE | 89.4 | 73.0 | 79.6 | 66.8 | 60.7 | 62.9 |
| ABAE-init | 90.1 | 76.7 | 82.9 | 68.5 | 63.3 | 65.4 |
| MATE | 87.8 | 79.4 | 83.4 | 68.1 | 63.8 | 65.6 |
| AE-CSA | 85.6 | 86.0 | 85.8 | – | – | – |
| CMAM | 83.2 | 82.9 | 83.1 | 70.0 | 70.4 | 70.1 |
| BARLAT | **90.8** | 84.6 | **87.5** | **74.9** | **71.7** | **72.4** |

Bold shows the numbers of best performing models against each of the metrics
The symbol '–' indicates the source code is not available in the respective papers. Hence, results are not reproduced

combines the sentence representation generation and short text clustering in an end-to-end learning fashion; (3) Adversarial training adopted in BARLAT can effectively improve clustering performance.

### 5.1 Ablation Study

An ablation study is performed to understand the effectiveness of the critical components of the proposed model. Table 7 reports the macro-averaged F1-score of different variations of BARLAT. The results show that DK-BERT, Redundancy regularizer, Seed word regularizer, and Adversarial training significantly contribute to improving performance. The contribution of domain knowledge BERT and adversarial training is slightly more than Redundancy regularizer and Seed word regularizer in the proposed network. Performance comparison of BARLAT and BARLAT (word) also explores the effect of cluster-level adversarial perturbations by comparing it with word-level adversarial perturbations. The results show that BARLAT performs slightly better than BARLAT (word), and hence, cluster-level adversarial perturbations are more effective than word-level adversarial perturbations.

### 5.2 Hyperparameter Analysis

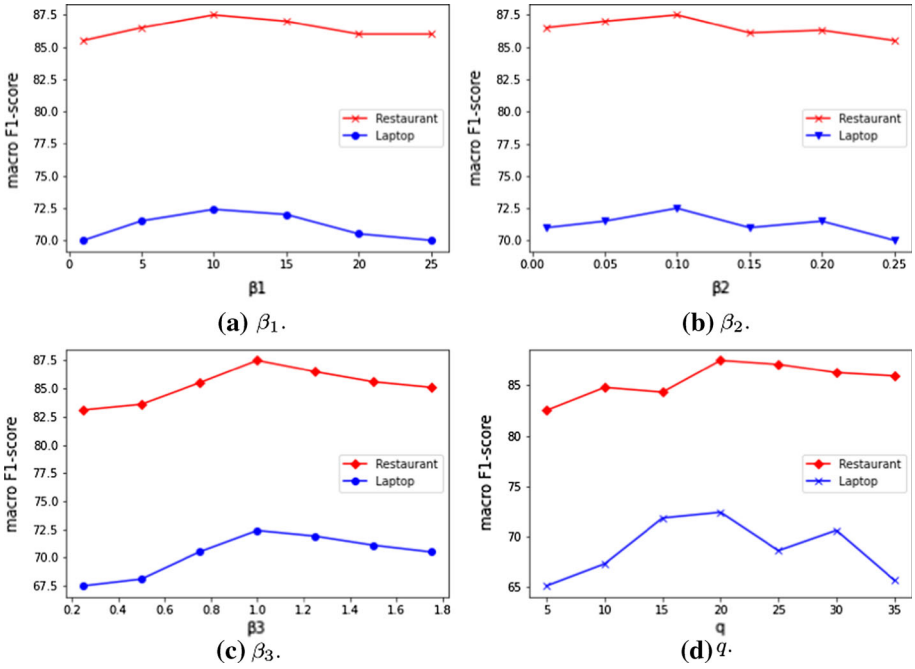
In this section importance of various hyperparameters is discussed.

– *Effect of* $\beta_1$, $\beta_2$, *and* $\beta_3$ The performance of BARLAT is measured by setting the range of values of one hyperparameter while keeping the values of others constant. The Fig. 3a–c shows how different values of $\beta_1$, $\beta_2$, and $\beta_3$ affects the performance of BARLAT.

**Table 7** Performance (weighted macro F1 score in %) comparisons of different version of BARLAT model on Restaurant & Laptop datasets

| Method | Restaurant | Laptop |
|---|---|---|
| BARLAT w/o adv & redundancy reg. & seed word reg. & DK-BERT | 79.3 | 63.8 |
| BARLAT w/o adv & redundancy reg. & seed word reg. | 81.5 | 66.3 |
| BARLAT w/o adv & redundancy | 82.7 | 66.9 |
| BARLAT w/o adv | 83.3 | 67.7 |
| BARLAT (word) | 86.1 | 70.3 |
| BARLAT | **87.5** | **72.4** |

Bold shows the numbers of best performing models against each of the metrics




**(a)** $\beta_1$. **(b)** $\beta_2$. **(c)** $\beta_3$. **(d)** $q$.

**Fig. 3** Performance of BARLAT with differnt values of $\beta_1$, $\beta_3$, $\beta_3$ and $q$

The best possible values of these hyperparameters are selected through this process as mentioned in Sect. 4.4.

– *Negative sample size (q)* The performance of BARLAT is measured by experimenting with different values of $q$ ranging from 5 to 35. Figure 3d shows how different values of q affects the performance of BARLAT. It is observed that BARLAT achieves the best performance when we set negative sample size of 20 as the optimal hyper-parameter for both Restaurant and Laptop datasets.
– *Importance of* $\epsilon$ The proposed model generates adversarial examples by adding perturbation to the cluster embedding. The hyperparameter $epsilon$ controls the magnitude of the perturbation. Different values of $\epsilon$ affect the loss function that leads to fluctuation in the performance of BARLAT. To better understand this fluctuation, the proposed

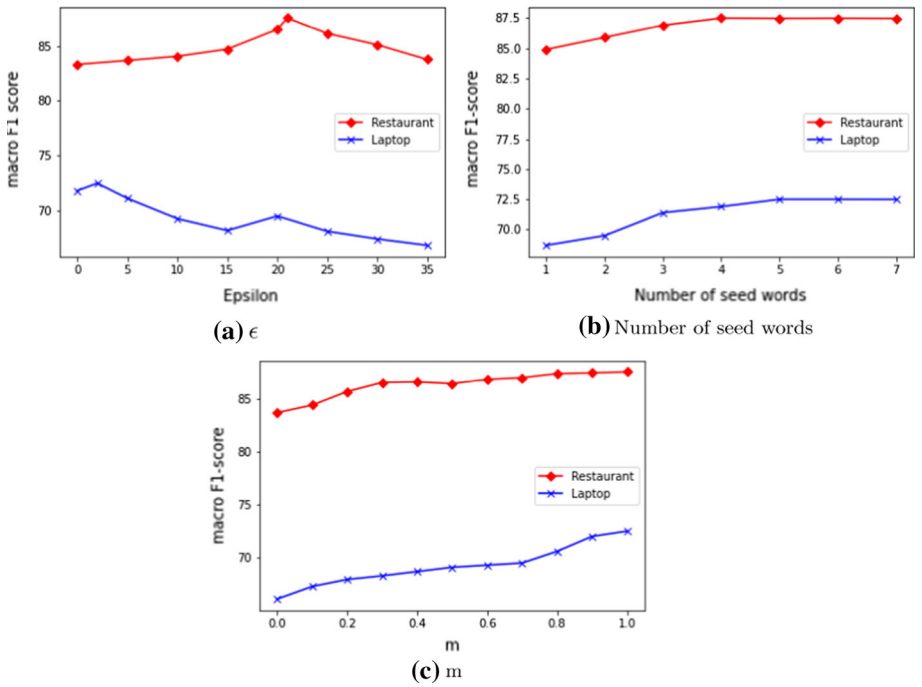



**Fig. 4** Performance of BARLAT with different values of $\epsilon$, number of seed words and $m$

model is built by taking different values of $\epsilon$ ranging from 1 to 35. Figure 4a shows the macro-averaged F1-score of BARLAT on Restaurant and Laptop datasets. It is observed that BARLAT achieves the best performance when $\epsilon$ is set to 21 and 2 for Restaurant and Laptop datasets, respectively.

– *Number of seed words* Seed words play an essential role in the training of the proposed model. To understand the importance of the same, we make a list of ten words for each aspect category from both datasets. We randomly pick $p$ seed words from each list, run BARLAT five times, and take the average macro-F1 scores. The values of $p$ are varied from 1 to 7 and the results of aspect category detection task are plotted in Fig. 4b. The results show that the BARLAT achieves the best performance by using less than five seed words for both the datasets. BARLAT achieves the best performance on Restaurant and Laptop datasets by using the number of seed words four and five, respectively. Subsequently, even we add more seed words, the performance of BARLAT gradually saturates on both the datasets. It shows that the proposed model needs only a small set of seed words to perform well.
– *Effect of margin (m)* Performance of BARLAT is measured by experimenting with different values of margin $m$ in Eq. 7. Figure 4c shows the macro F1-score of BARLAT on Restaurant and Laptop datasets on different values of $m$ ranging from 0 to 1. The results show that the performance of BARLAT improves when $m$ is set to values around 1 in both restaurant and laptop datasets.

**Table 8** Case analysis: Input sentences with predicted aspect category by various models

| Input sentence | ABAE | BARLAT w/o Adv & redundancy reg & seed word reg. & DK-BERT | BARLAT w/o Adv | BARLAT |
|---|---|---|---|---|
| We only ordered desserts and drinks, but no refills were offered. (**Service**) | Food | Service | Service | Service |
| Worse of all, $ 60 was erroneously added to our $ 80 bill. (**Service**) | Price | Service | Service | Service |
| Really though, where's the seasoning ?. (**Food**) | Ambience | Miscellaneous | Food | Food |
| Great spot, whether looking for a couple of drinks or quiet dinner. (**Ambience**) | Miscellaneous | Food | Ambience | Ambience |
| It's a pleasure to type on. (**Keyboard**) | Software | Support | Keyboard | Keyboard |
| Of course, it is crowded but who cares.(**Ambience**) | Miscellaneous | Food | Miscellaneous | Ambience |

## 5.3 Case Study

A case study is presented by taking some examples from Restaurant and Laptop datasets to demonstrate the effectiveness of the proposed model. Table 8 shows the details of the predicted aspect category corresponding to taken examples. In the first sentence, ABAE pays more attention to "desserts and drinks," which is a food item, and predicts aspect category as *Food*. However, other models *BARLAT w/o Adv & redundancy reg. & seed word reg. & DK-BERT*, *BARLAT w/o & Adv* and *BARLAT*, which use contextual word embedding (BERT) for generating the sentence representation predict the correct aspect category as *Service*. This shows the importance of contextual word embedding (BERT) and sentence clustering through cluster-level attention, enabling models to predict the correct aspect category. Similarly, the second sentence also quantifies the importance of BERT and attentive representation learning, where ABAE makes wrong aspect category prediction as *Price* by focusing more on words like "bill". Still, variants of BARLAT models effectively understand the overall meaning of the sentence and correctly predict the aspect category as *Service*. Analysis of third, fourth, and fifth sentences is very interesting, where *BARLAT w/o Adv & redundancy reg. & seed word reg. & DK-BERT*, which uses $BERT_{BASE}$ fails to understand the overall meaning of the sentence and makes the wrong prediction like *ABAE*. However, *BARLAT w/o Adv* and *BARLAT*, which explores domain-specific knowledge using DK-BERT, correctly identifies the aspect categories in both the sentences. It illustrates the importance of DK-BERT, which encodes the sentence representation more effectively by using domain knowledge. The sixth sentence illustrates the importance of adversarial training wherein all models predict the wrong aspect category except *BARLAT*. It shows a combination of DK-BERT and adversarial training helps in learning the cluster embedding in a better way and eventually enhances the performance of *BARLAT*.

### 5.4 Error Analysis

The errors of BARLAT are classified into the following categories:

- *Sentence without any context* Some review sentences are very short and use opinion words without any context. The proposed model is unable to detect the correct aspect category in such sentences. For example, in the sentence, *Either way highly annoying.*, target of the opinion term *annoying* is not clear.
- *Expressing an opinion about an aspect category in the midst of other experiences* In some review sentences, users are expressing an opinion about an aspect category while describing their experiences, which makes aspect detection challenging. Examples include *We came across this restaurant by accident while at a DUMBO art festival and thoroughly enjoyed our meal.*. Here, opinion about the meal (Food) is expressed while describing an event.
- *Ambiguous use of opinion* Few review sentences express an opinion in an ambiguous way for more than one target. For example, in the sentence *I really enjoyed my meal here.*, it is not very clear that user is expressing an opinion about *meal* or *here* (place). Likewise, in the review sentence *i highly recommend this place to all that want to try indian food for the first time.*, it is very hard for the model to understand whether aspect category is *this place* or *indian food*.

## 6 Conclusion and Future Work

In this paper, a novel model BARLAT is introduced for aspect category detection with minimal guidance from users. It integrates the domain-specific BERT-based sentence representation with clustering in a unified framework through cluster-level attention. Adversarial perturbations are further added to cluster embeddings, which enhances the performance of the model. The extensive experiments on two real-life datasets show that BARLAT outperforms the state-of-the-art models for aspect category detection. The detailed analysis of the proposed BARLAT model shows the contributions of its constituent components. Future work will focus on low-resource non-English languages, as getting labeled data for such languages is challenging. The multilingual masked language models (MLM) like mBERT [16], and XLM-R [44] will be used to investigate how cross-lingual transfer helps to solve aspect category identification task in multilingual settings.

## Declarations

**Conflict of interest** The authors have no relevant financial or non-financial interests to disclose. The authors have no conflicts of interest to declare that are relevant to the content of this article. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no financial or proprietary interests in any material discussed in this article.

## References

1. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 168–177

2. Liu B, Zhang L (2012) A survey of opinion mining and sentiment analysis. In: Mining text data. Springer, pp 415–463
3. Pontiki M, Galanis D, Papageorgiou H, Androutsopoulos I, Manandhar S, Al-Smadi M, Al-Ayyoub M, Zhao Y, Qin B, De Clercq O, et al. (2016)Semeval-2016 task 5: Aspect based sentiment analysis. In: *10th international workshop on semantic evaluation (SemEval 2016)*
4. Qiu G, Liu B, Jiajun B, Chen C (2011) Opinion word expansion and target extraction through double propagation. Comput Linguist 37(1):9–27
5. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
6. Brody S, Elhadad N (201) An unsupervised aspect-sentiment model for online reviews. In: Human language technologies: the 2010 annual conference of the North American chapter of the association for computational linguistics. Association for Computational Linguistics, pp 804–812
7. Zhao WX, Jiang J, Yan H, Li X (2010) Jointly modeling aspects and opinions with a maxent-lda hybrid. In: *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics, pp 56–65
8. Chen Z, Mukherjee A, Liu B (2014) Aspect extraction with automated prior knowledge learning. In: *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: long papers)*, pp 347–358
9. Mimno D, Wallach H, Talley E, Leenders M, McCallum A (2011) Optimizing semantic coherence in topic models. In: Proceedings of the 2011 conference on empirical methods in natural language processing, pp 262–272
10. Miao Y, Yu L, Blunsom P (2016) Neural variational inference for text processing. In: International conference on machine learning, pp 1727–1736
11. Srivastava A, Sutton C (2017) Autoencoding variational inference for topic models. Preprint arXiv:1703.01488
12. Liao M, Li J, Zhang H, Wang L, Wu X, Wong K-F (2019) Coupling global and local context for unsupervised aspect extraction. In: *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp 4571–4581
13. He R, Lee WS, Ng HT, Dahlmeier D (2017) An unsupervised neural attention model for aspect extraction. In: Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers), pp 388–397
14. Tulkens S, van Cranenburgh A (2020) Embarrassingly simple unsupervised aspect extraction. Preprint arXiv:2004.13580
15. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. Preprint arXiv:1301.3781
16. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. Preprint arXiv:1810.04805
17. Titov I, McDonald R (2008) Modeling online reviews with multi-grain topic models. In: Proceedings of the 17th international conference on World Wide Web, pp 111–120
18. Mukherjee A, Liu B (2012) Aspect extraction through semi-supervised modeling. In: Proceedings of the 50th annual meeting of the association for computational linguistics: Long papers-volume 1. Association for Computational Linguistics, pp 339–348
19. Yan X, Guo J, Lan Y, Cheng X (2013) A biterm topic model for short texts. In: Proceedings of the 22nd international conference on World Wide Web, pp 1445–1456
20. Zbontar J, Jing L, Misra I, LeCun Y, Deny S (2021) Barlow twins: self-supervised learning via redundancy reduction
21. Lara JS, González FA (2020) Dissimilarity mixture autoencoder for deep clustering. Preprint arXiv:2006.08177
22. Hadifar A, Sterckx L, Demeester T, Develder C (2019) A self-training approach for short text clustering. In: Proceedings of the 4th workshop on representation learning for NLP (RepL4NLP-2019), pp 194–199
23. Zhang W, Dong C, Yin J, Wang J (2019)Attentive representation learning with adversarial training for short text clustering. Preprint arXiv:1912.03720
24. García-Pablos A, Cuadros M, Rigau G (2018) W2vlda: almost unsupervised system for aspect based sentiment analysis. Exp Syst Appl 91:127–137
25. Angelidis S, Lapata M (2018) Summarizing opinions: aspect extraction meets sentiment prediction and they are both weakly supervised. Preprint arXiv:1808.08858
26. Sokhin T, Khodorchenko M, Butakov N (2020) Unsupervised neural aspect search with related terms extraction. Preprint arXiv:2005.02771
27. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2013) Intriguing properties of neural networks. Preprint arXiv:1312.6199

28. Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. Preprint arXiv:1412.6572
29. Wu Y, Schuster M, Chen Z, Le QV, Norouzi M, Macherey W, Krikun M, Cao Y, Gao Q, Macherey K, et al. (2016) Google's neural machine translation system: bridging the gap between human and machine translation. Preprint arXiv:1609.08144
30. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: *Advances in neural information processing systems*, pp 5998–6008
31. Xiao H (2018) bert-as-service. https://github.com/hanxiao/bert-as-service
32. Socher R, Karpathy A, Le QV, Manning CD, Ng AY (2014) Grounded compositional semantics for finding and describing images with sentences. Trans Assoc Comput Linguist 2:207–218
33. Weston J, Bengio S, Usunier N (2011) Wsabie: scaling up to large vocabulary image annotation. In: Proceedings of the twenty-second international joint conference on artificial intelligence—volume volume three, IJCAI'11. AAAI Press, pp 2764–2770
34. Iyyer M, Guha A, Chaturvedi S, Boyd-Graber J, Daumé III H (2016) Feuding families and former Friends: unsupervised learning for dynamic fictional relationships. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, San Diego, California, June . Association for Computational Linguistics, pp 1534–1544
35. Miyato T, Maeda S-i, Koyama M, Nakae K, Ishii S (2015) Distributional smoothing with virtual adversarial training. Preprint arXiv:1507.00677
36. Wang L, Liu K, Cao Z, Zhao J, De Melo G (2015)Sentiment-aspect extraction based on restricted boltzmann machines. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: long papers), pp 616–625
37. He R, McAuley J (2016) Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In: proceedings of the 25th international conference on world wide web, pp 507–517
38. Li N, Chow C-Y, Zhang J-D (2019) Seeded-btm: enabling biterm topic model with seeds for product aspect mining. In: *2019 IEEE 21st international conference on high performance computing and communications; IEEE 17th international conference on smart city; IEEE 5th international conference on data science and systems (HPCC/SmartCity/DSS)*. IEEE, pp 2751–2758
39. Goldberg Y, Levy O (2014) word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. Preprint arXiv:1402.3722
40. Vargas DS, Pessutto LRC, Moreira VP (2020)Simple unsupervised similarity-based aspect extraction. Preprint arXiv:2008.10820
41. Luo L, Ao X, Song Y, Li J, Yang X, He Q, Yu D (2019)Unsupervised neural aspect extraction with sememes. In: IJCAI, pp 5123–5129
42. Xu H, Liu B, Shu L, Yu PS (2019) Bert post-training for review reading comprehension and aspect-based sentiment analysis. Preprint arXiv:1904.02232
43. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. Preprint arXiv:1412.6980
44. Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, Grave E, Ott M, Zettlemoyer L, Stoyanov V (2019) Unsupervised cross-lingual representation learning at scale. Preprint arXiv:1911.02116