



Cross-domain aspect-based sentiment analysis using domain adversarial training

Joris Knoester¹ · Flavius Frasincar¹ · Maria Mihaela Truşcă²

Received: 13 March 2023 / Revised: 27 September 2023 / Accepted: 1 October 2023 /
Published online: 22 November 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

Over the last decades, the increasing popularity of the Web came together with an extremely large volume of reviews on products and services useful for both companies and customers to adjust their behaviour with respect to the expressed opinions. Given this growth, Aspect-Based Sentiment Analysis (ABSA) has turned out to be an important tool required to understand people's preferences. However, despite the large volume of data, the lack of data annotations restricts the supervised ABSA analysis to only a limited number of domains. To tackle this problem a transfer learning strategy is implemented by extending the state-of-the-art LCR-Rot-hop++ model for ABSA with the methodology of Domain Adversarial Training (DAT). The output is a cross-domain deep learning structure, called DAT-LCR-Rot-hop++. The major advantage of DAT-LCR-Rot-hop++ is the fact that it does not require any labeled target domain data. The results are obtained for six different domain combinations with testing accuracies ranging from 35% up until 74%, showing both the limitations and benefits of this approach. Once DAT-LCR-Rot-hop++ is able to find the similarities between domains, it produces good results. However, if the domains are too distant, it is not capable of generating domain-invariant features. This result is amplified by our additional analysis to add the neutral aspects to the positive or negative class. The performance of DAT-LCR-Rot-hop++ is very dependent on the similarity between distributions of source and target domain and the presence of a dominant sentiment class in the training set.

This article belongs to the Topical Collection: *Special Issue on Web Information Systems Engineering 2022*
Guest Editors: Richard Chbeir, Helen Huang, Yannis Manolopoulos and Fabrizio Silvestri

✉ Maria Mihaela Truşcă
maria.trusca@csie.ase.ro

Joris Knoester
joris.knoester@gmail.com

Flavius Frasincar
frasincar@ese.eur.nl

¹ Department of Econometrics, Erasmus University Rotterdam, Burgemeester Oudlaan 50, Rotterdam 3062 PA, the Netherlands

² Department and Informatics and Economic Cybernetics, Bucharest University of Economic Studies, Piata Romana, Bucharest 010374, Romania

Keywords Aspect-based sentiment classification · Transfer learning · Adversarial training

1 Introduction

The increasing Web popularity for digital communication has started with the first social media platform, Six Degrees in 1997 [10] and continued until nowadays, when over 3.7 billion people, which accounts for 48% of the world population, are exchanging information on the social Web [30]. Due to this popularity and the subsequent increasing amount of reviews on products or services, a wealth of useful information is available to businesses and consumers alike. But due to this increased amount of reviews available, it is very difficult or even impossible to analyse these by hand.

Aspect-Based Sentiment Analysis (ABSA) classifies a person’s feeling towards specific aspects [31]. ABSA is concerned with target extraction (TE), aspect detection (AD), and target sentiment classification (SC) [27]. This paper concentrates on the last task, SC, classifying the user’s sentiment with respect to the aspects, and is called Aspect-Based Sentiment Classification (ABSC) [2]. As an example, we provide the sentence “the atmosphere and service were terrible, but the food was good”. In this case, the “atmosphere”, “service”, and “food” are all *target words* and “terrible” and “good” are the expressions that give *context* to these target words to be able to label the targets as positive, neutral, or negative. As one can notice, the context around the target words is essential to capture the explicit aspect sentiments.

There are multiple practical applications of ABSC. By evaluating and deciding which features need improvement, a company can apply specific enhancements to their products, efficiently improving their customer services. At the same time, social media platforms such as Facebook and Twitter can implement ABSC on tweets and messages, and sell this valuable information to the marketing department of multinationals. In addition, financial firms can apply ABSC to forecast the feelings of financial individuals towards the economic market and thereby predict future stock movements. This would have been extremely beneficial for Melvin Capital before the whole GameStop phenomena at Reddit and might have prevented enormous losses [3]. Last, knowing the opinion of previous customers can help potential clients make better-informed buying decisions.

An issue that has gained much attention recently is the limited availability of labeled data. Generating new labeled data for specific domains is expensive, time-consuming, and requires manual labour. In order to decrease the dependence on labeled data, transfer learning, also called cross-domain learning, is a valuable solution [21]. This approach concentrates on training a model on a related source domain and then predicting for a different target domain. Several state-of-the-art cross-domain models rely on Domain-Adversarial Neural Networks (DANN) introduced in [8]. These neural networks are applied to diverse tasks, ranging from textual entailment analysis [14] to image classification [41]. However, little research is available on ABSA using Domain-Adversarial Training (DAT). To our knowledge, there is limited research on training domain-invariant features for Aspect-Based Sentiment Classification (ABSC) using DANN. Prior methods for sentiment classification apply a Support Vector Machine (SVM) [22] in order to predict the sentiments, but due to its shortcomings, this system was replaced by knowledge-based models [29] and deep learning algorithms [16]. Whereas the deep learning methods are flexible, knowledge-based models require more manual labour, but achieve better results [19]. Since a combination of both approaches benefits from the advantages of both solutions, several researchers merge the methods into a hybrid model [32, 35].

The proposed methodology of this paper is based on the hybrid HAABSA++ model proposed in [33], a state-of-the-art ABSC approach that produces excellent results for the commonly used SemEval 2015 [24] and 2016 [25] data sets, attaining a classification accuracy of 81.7% and 87.0%, respectively. Nevertheless, our study focuses on the development of a neural network that can classify texts on multiple different domains. For this reason, the knowledge-based ontology part of the HAABSA++ approach is eliminated, leaving the LCR-Rot-hop++ model based on [44]. This model is expanded by applying the structure of DAT as proposed by [8]. More specifically, the neural network is trained concurrently on labeled instances of a source domain and unlabeled instances from a target domain. We call the newly established system Domain Adversarial Training LCR-Rot-hop++, abbreviated as DAT-LCR-Rot-hop++. All source code and data can be retrieved from <https://github.com/jorisknoester/DAT-LCR-Rot-hop-PLUS-PLUS>.

The current work is an extension of our previous work [15] dedicated to the DANN applicability as a knowledge transfer method for ABSA domains without annotations. Additionally to [15], we explore how sensitive our results are with respect to the discriminator architecture, finding out that a too simple structure will mainly facilitate the prediction of the majority class. We also perform additional experiments on how the distribution of the training data affects the DANN performance. As expected, a DAT learning strategy requires a substantial amount of data for all class labels in order to avoid majority voting.

The rest of this paper is structured as follows. First, Section 2 gives the relevant literature concerned with ABSC, the different components of the LCR-Rot-hop++ model, and the ideas behind transfer learning. Second, Section 3 provides a short description of the data together with a couple of descriptive statistics. Third, Section 4 concentrates on the methodology of this research for which the results are reported in Section 5. Last, Section 6 provides our conclusion and suggestions for future work.

2 Related work

ABSC is concerned with classifying a person's sentiment towards specific aspects in a sentence. One of the first works on sentiment analysis is presented in [23] and sentiment analysis has been a hot topic ever since. While the traditional methods were mainly developed based on knowledge-based systems providing higher prediction scores for domain-specific documents [19], the more recent ones tend to rely more on deep learning solutions. Despite being considered alternative solutions, in [38] it is observed that the two approaches are in fact complementary. Consecutively, a hybrid method which incorporates both the domain ontology and a deep learning neural network is introduced in [28]. After several additional improvements, this has resulted in the HAABSA++ method [33].

HAABSA++ method has two steps so that if the ontology is unable to provide reliable results, the LCR-Rot-hop++ neural network is used as a backup. As the topic of our work is transfer learning, we focus only on the neural network. LCR-Rot-hop++ is actually a bi-directional Long Short-Term Memory (Bi-LSTM) model accompanied by attention. This attention layer is able to put focus on specific parts of a sentence, thereby limiting the influence of the less important words. On top of this attention layer, a hierarchical attention layer is employed, enabling the model to process the text on the sentence-level, bringing together the local sentence representations. The last component of the LCR-Rot-hop++ model is its rotary system that increases the interaction between targets and contexts by sharing information in order to capture the most indicative sentiment words.

Transfer learning [21] is a machine learning technique that focuses on storing information from one data set and applying this knowledge on another. Because obtaining annotated data is costly and time-consuming, new models must be developed to provide reliable results for multiple domains. The variety of methods of transfer learning is continuously expanding. One of the proposed solutions focuses on freezing the first layers of an LSTM neural network [4]. This approach is based on the fact that the source domain contains valuable universal information, and higher layer neurons tend to specialize more towards the target domain, while the lower hidden layers generate more common word features [39]. A state-of-the-art method, BertMasker [40] uses the idea of masking [6] for their BERT Base network. BertMasker is able to mask domain-related words. This transforms the remaining sentence text to be domain-invariant, while maintaining its most sentiment-explicit words. Another solution scientists developed is a domain adapting network by creating counterfactual features [13]. These counterfactual depictions reduce the inductive bias of the source domain. The designed positive (negative) counterfactuals bridge the dimensional gap between the positive (negative) classified instances of the source and target domain. Different from the previous works, in this research we apply the methodology of Generative Adversarial Networks (GAN). This last solution is introduced in [9] and has shown superior performance in a broad range of scientific areas, such as image classification [5], event detection [12], and textual cross-domain sentiment classification [43]. Applying the logic of GAN, DANN is introduced in [7]. A DANN model is able to perform machine learning tasks on unlabeled target domain data, while trained on a labeled source domain with a relatively similar distribution, both in terms of polarity distribution and batch size [7]. The advantage that one does not need annotated target data makes DANN very valuable for future cross-domain deep learning problems and is therefore an important contribution to the existing machine learning techniques.

To implement DANN for cross-domain sentiment classification, we rely on the features computed by the LCR-Rot-hop++ model. A similar study to ours based on adversarial learning is discussed also in [37]. However, in [37], the class and domain discrimination use features computed by different models. As opposed, in our work, we consider that class and domain discrimination are complementary tasks that will take advantage of using shared features computed by the LCR-Rot-hop++ model.

3 Data

In this paper, two different data sets are used. These are the Semantic Evaluation (SemEval) 2014 [26], and the Amazon/LibraryThing (ALT) 2019 [18]. SemEval 2014 includes information about the restaurant and laptop domain, and ALT contains the data for the book domain. The reviews of the data sets are divided into single sentences, consisting of one or more aspects and sentimental context words, which are used to classify the polarities (positive, neutral, or negative). The partitioning of the data into a training and test set is done as follows. The aspects are divided into 80% training and 20% testing. The training set consists of 80% pure training and 20% validation to compute the optimal values for the hyperparameters. The results of the split into training and test data are presented in Table 1.

In terms of data pre-processing, the same approach is applied as introduced in [36] and [44]. The implicitly opinionated review sentences contain a sentiment, but the aspect term is missing. This makes it impossible to perform ABSC using the LCR-Rot-hop++ model. In addition to this, it could occur that an aspect has conflicting sentiments. This happens when there is both negative and positive context towards an aspect. Both the conflicting sentiment and the implicitly opinionated sentences are removed from the datasets.

Table 1 The distribution of the aspect sentiment polarities of the three domains

	Restaurant		Laptop		Book	
	Train	Test	Train	Test	Train	Test
Positive	60.1%	65.1%	42.7%	52.4%	25.8%	32.3%
Neutral	17.7%	17.5%	19.8%	26.2%	63.1%	57.1%
Negative	22.2%	17.4%	37.5%	21.4%	11.1%	10.6%
Total	3600	1122	2250	701	2700	804

DAT-LCR-Rot-hop++ requires aspects from two domains to go through the model. These domains are defined as the source domain and the target domain. Only the source domain instances have a sentiment label attached to them. As a result, during training, the aspects of the source domain consist of two labels, the domain class d , and the sentiment category y , while the instances of the target domain only contain a domain class d . For testing, the polarity labels of the target domain aspects are determined in order to evaluate the performance of DAT-LCR-Rot-hop++. Our proposed model is concurrently trained on one source and one target domain to obtain domain-invariant features. Results are presented for six different domain combinations (restaurant-laptop, restaurant-book, laptop-restaurant, laptop-book, book-restaurant, and book-laptop). As an example, the restaurant-laptop model means that the restaurant data set is the source domain, and the laptop data set is the target domain. When training is finished, the test instances of the target domain are fed into the model for evaluation. The performance of DAT-LCR-Rot-hop++ is analysed according to its predicting sentiment accuracy of the target test aspects.

4 Framework

A GAN model [9] generally consists of two additional elements on top of the neural feature extractor, which is the LCR-Rot-hop++ model in this paper. The feature extractor transforms an input sentence into a vector representation that is ought to capture the important characteristics of the sentence. The other two extra elements are the generator and discriminator. A visual representation of our neural network is shown in Figure 1.

The DAT method introduced in [7] is able to adapt to target domains without any labeled target data. This is done by generating deep features that are discriminative for the main learning classifying task by using the labeled sentiments of the source domain. This method ensures that these representations are invariant to shifts between the source and target domain in order to be domain-in-discriminative by using the domain class of both the source and target domain. The proposed DANN solution is revisited in [8], which provides a more detailed and elaborate description of the mathematics behind the system.

The change to the GAN network in order to conform to a DANN model means replacing the generator with a Gradient Reversal Layer (GRL). GRL aims to make the task of the domain discriminator as hard as possible, which is the direct connection to GAN. In DAT-LCR-Rot-hop++, the loss of the domain discriminator is passed through GRL, which reverses the gradient before back-propagation into the feature extractor. This causes the hidden layers of LCR-Rot-hop++ to react by constructing features which will not be recognised as a certain domain by the domain discriminator. This process continues until at some point, the word vectors are completely domain-invariant, which causes the domain discriminator to be unable of distinguishing the source and target domain in the shared feature representations. The rest

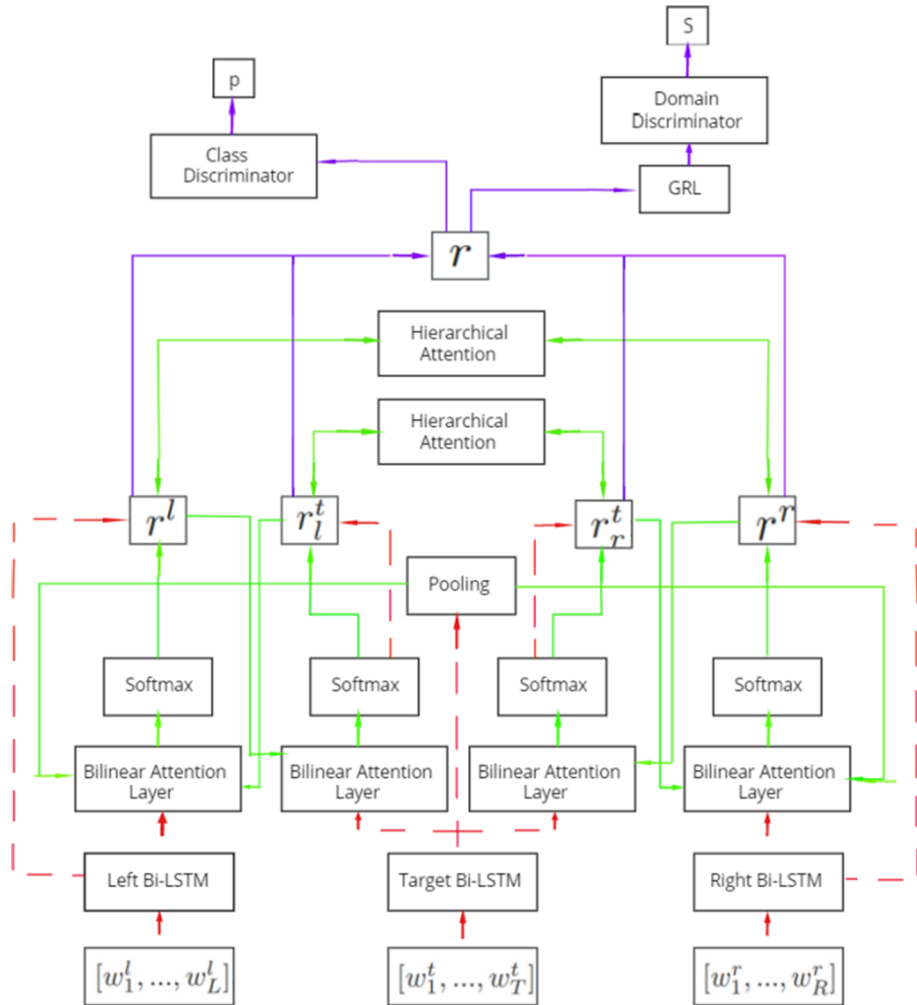


Figure 1 A visualisation of the LCR-Rot-hop++ model

of this section is structured as follows. Section 4.1 presents the feature extractor LCR-Rot-hop++, Section 4.2 describes the structure of our DANN proposed model, and Section 4.3 presents the training procedure for our model.

4.1 LCR-Rot-hop++

The LCR-Rot model is introduced in [44], extended in [34] with multi-hop attention to LCR-Rot-hop, and further expanded with deep contextual word embeddings and hierarchical attention resulting in LCR-Rot-hop++, as described in [33].

First, the sentences are split into three separate parts, consisting of the left context, $[s_1^l, \dots, s_L^l]$, target phrase, $[s_1^t, \dots, s_T^t]$, and right context, $[s_1^r, \dots, s_R^r]$. These sentence fractions have lengths L, T, and R, respectively, such that $L+T+R$ is equal to the complete sentence

length, S . These chunks are converted to contextual word embeddings using the pre-trained BERT Base model ($L = 12, A = 12, H = 768$) as introduced in [6]. The final word embeddings are calculated by summing the last 4 layers of the BERT model:

$$BERT_i = \sum_{j=9}^{12} H_{i,j}. \tag{1}$$

All word embeddings have a dimension of $1 \times d$, where d is equal to $H = 768$. So, the word representations are each a vector of size 768.

Next, the left context word embeddings, $[w_1^l, \dots, w_L^l]$, the target word embedding, $[w_1^t, \dots, w_T^t]$, and the right word embedding, $[w_1^r, \dots, w_R^r]$, are each the input for a three hidden layer bi-LSTM feed-forward neural network, resulting in the hidden states $[h_1^l, \dots, h_L^l]$, $[h_1^t, \dots, h_T^t]$, and $[h_1^r, \dots, h_R^r]$. These hidden states all have dimension of $2d \times 1$ due to the bidirectional structure. This process is shown by the dark red dashed arrows in Figure 1.

Afterwards, a rotary attention mechanism is applied to the outputs to capture the most indicative words in the left and right contexts and the target phrase. This is a two-step mechanism. In the first step the target2context vectors are computed. This is done by average pooling the target phrase, which results in r^t , as shown in (2).

$$r_{2d \times 1}^t = \text{pooling}([\underset{2d \times 1}{h_1^t}, \dots, \underset{2d \times 1}{h_T^t}]) \tag{2}$$

Then, the neural network utilises this as extra input in the bilinear attention layer of the two context parts of the sentence. In this attention layer, the target phrase representation is combined with the h_i^l and h_i^r for the left and right contexts, respectively. A bilinear attention score f , see (3), is employed to achieve accurate representations of the left and right contexts. In the remainder of this section, the left context representation will be used as example to avoid duplicity.

$$f(h_i^l, r^t) = \tanh(\underset{1 \times 1}{h_i^l} \times \underset{1 \times 2d}{W_c^l} \times \underset{2d \times 2d}{r^t} + \underset{1 \times 1}{b_c^l}), \tag{3}$$

where h_i^l is the hidden state of the left context bi-LSTM, W_c^l represents the weight matrix, and b_c^l depicts the bias term, for $i = 1, \dots, L$.

Next, the attention scores are normalised to range from 0 to 1 by a softmax function, which results in α_i^l . This is defined as follows:

$$\alpha_i^l = \frac{\exp(f(h_i^l, r^t))}{\sum_{j=1}^L \exp(f(h_j^l, r^t))}. \tag{4}$$

Last, the left and right context representation can be retrieved by computing a weighted combination of the hidden states and the attention scores:

$$r_{2d \times 1}^l = \sum_{i=1}^L \underset{1 \times 1}{\alpha_i^l} \times \underset{2d \times 1}{h_i^l}. \tag{5}$$

Next, in the second step of the rotary system, these left and right context representations, called r^l and r^r , are fed into the bilinear attention layer of the left- and right-aware representations of the target phrase. Now, the same methodology is applied as described for the previous context depictions applied to r^l or r^r and h_i^t . This results in the context2target vectors, r_i^l and r_i^r :

$$r_{2d \times 1}^l = \sum_{i=1}^T \underset{1 \times 1}{\alpha_i^l} \times \underset{2d \times 1}{h_i^t}, \tag{6}$$

where h_i^t represents the hidden states of the target phrase bi-LSTM layers and $\alpha_i^{t_l}$ is the attention score associated with the target phrase’s hidden output with respect to the left context.

Both representative target phrase features, r_l^t and r_r^t , are then used as input for the first step, the target2context computation. As a result, the average pooling in (2) is skipped, because the newly calculated r_l^t and r_r^t are the input to (3)–(5). This whole procedure is repeated three times as was decided to be optimal in [34]. It is shown by the light green arrows (light grey for black and white printing) in Figure 1.

After having completed the first rotary attention mechanism, the four representations are fed into a hierarchical attention system. This component helps overcome the issue of only utilising local information. First, the word features are split into two groups: the target representations, r_l^t and r_r^t , and the context representations, r^l and r^r . Both combinations are then separately fed into a new attention layer with attention score f :

$$f(v^i) = \tanh\left(v^i_{1 \times 2d} \times W_h^c + b_h^c \right), \tag{7}$$

1×1 $1 \times 2d$ $2d \times 1$ 1×1

where W_h^c is the weight matrix of the hierarchical layer for the current partition and b_h^c represents the bias term for the current partition. In the context case, $v^i \in \{r^l, r^r\}$, but for the target phrase $v^i \in \{r_l^t, r_r^t\}$.

As for the bilinear attention layer, the function value is normalised by (8):

$$\alpha^i = \frac{\exp(f(v^i))}{\exp(f(v^{i1})) + \exp(f(v^{i2}))}, \tag{8}$$

after which the representations are updated:

$$v_{2d \times 1}^i = \alpha^i_{1 \times 1} \times v_{2d \times 1}^i, \tag{9}$$

This procedure is also rerun multiple times, which is visualised by the green arcs (dark grey for black and white printing). According to the standard implementation of LCR-Rot-hop++, the four vectors are concatenated into one single vector, $r = r^l; r_l^t; r_r^t; r^r$ with dimensions $8d \times 1$ and then passed into a Multi-Layer Perceptron (MLP), which uses a softmax function to predict the polarities. Since in the current work we want to integrate the model into a DANN, the MLP layer is replaced by a domain adversarial component discussed in the next section.

4.2 DANN

The domain adversarial component is represented by two standard feed-forward MLPs, which are the class discriminator and the domain discriminator that takes as input the context and target representations concatenated into r . First, the domain discriminator aims to correctly classify the domain of r . The predicted domain is given by s . Classifying the domain is a binary problem with $s = 0$ for the source and $s = 1$ for the target domain labels. Next, the class discriminator uses a softmax function to compute the probabilities of the sentiment of the aspect, resulting in a 1×3 output vector, p . The polarity with largest probability will be chosen as the final sentiment. The sigmoid function is used for domain prediction because it shows good performance for examining binary cases and is applied by multiple researches in domain discriminators [11, 42]. The DAT component is visualised by the dark purple solid arrows in Figure 1.

The objective is reducing the error term of both the domain discriminator, denoted as $L_d(\theta_f, \theta_d)$, and the class discriminator (sentiment discriminator), denoted as $L_c(\theta_f, \theta_c)$. Here, θ represents the parameters of the feature extractor (LCR-Rot-hop++ without original MLP), the domain discriminator, and the class discriminator, defined by the underscores f , d , and c , respectively. Hence, the objective function to optimise is:

$$\min_{\theta} L_{c,d}(\theta_f, \theta_c, \theta_d) = L_d(\theta_f, \theta_d) + L_c(\theta_f, \theta_c). \quad (10)$$

However, as previously described, the GRL tries to fool the domain discriminator. After the domain is predicted and its parameters θ_d , are updated, the loss is back-propagated into the feature extractor to change the weights accordingly. But this loss first passes through the GRL, which reverses the gradient by multiplying it with $-\lambda$ in order to hinder the performance of the domain discriminator. The reversing of the gradient forces the hidden layers of the LCR-Rot-hop++ to respond by adjusting their weights in the exact opposite way as desired by the domain discriminator, hereby making the task of the domain classifier more difficult. As a result, the features become more domain indiscriminative. This process leads to the following adjusted loss function:

$$\min_{\theta} L'_{c,d}(\theta_f, \theta_c, \theta_d) = -\lambda L_d(\theta_f, \theta_d) + L_c(\theta_f, \theta_c), \quad (11)$$

$$L_d(\theta_f, \theta_d) = -\sum_{i=1}^N d_i * \log(s_i) + \pi_d * \|\theta_d\|^2, \quad (12)$$

$$L_c(\theta_f, \theta_c) = -\sum_{i=1}^n y_i * \log(p_i) + \pi_c * (\|\theta_f\|^2 + \|\theta_c\|^2). \quad (13)$$

Here d_i refers to the actual domain class, and y_i represents the real polarity. s_i is the predicted domain and p_i is the predicted sentiment. π represents the L2-regularisation term for the class and domain discriminator with underscore c and d , respectively. Last, n equals the source domain sample size, and N is the total sample size of the source and target domain data combined. As described, both the source and target aspects are fed into the domain discriminator, while only the source instances are passed into the class discriminator. Applying the principle of adversarial training, first the loss involving only the domain discriminator is maximised in order to learn the discriminator to differentiate the domains. This min-max situation resolves to:

$$\hat{\theta}_d = \operatorname{argmax}_{\theta_d} L'_{c,d}(\hat{\theta}_f, \hat{\theta}_c, \theta_d) \quad (14)$$

$$(\hat{\theta}_f, \hat{\theta}_c) = \operatorname{argmin}_{\theta_f, \theta_c} L'_{c,d}(\theta_f, \theta_c, \hat{\theta}_d) \quad (15)$$

At this saddle point, the parameters of the domain discriminator, θ_d (12), minimise the domain classification error. Secondly, θ_c and θ_f are computed to optimise (11) by minimising the sentiment prediction loss and maximising the domain classification error (confusing the discriminator of domains). The hyperparameter λ regulates the balance and trade-off between both goals.

The original DANN paper [7] implements Stochastic Gradient Descent (SGD) optimisation. However, the state-of-the-art image classifying model proposed in [20] shows that utilising the faster momentum method [17] instead of SGD also produces accurate results. In each iteration, the parameters of the neural network will be updated according to this method:

$$v_t \leftarrow \gamma * v_{t-1} + \eta * \nabla_{\theta_k} L(\theta_k) \quad (16)$$

$$\theta_k \leftarrow \theta_k - v_t. \quad (17)$$

Here, the hyperparameters are the learning rate, η , and the momentum factor, γ . In addition, the parameter θ_k represents the weights and biases for the domain discriminator, the feature extractor, and the class discriminator, with $k = d$, $k = f$, and $k = c$, respectively. Last, L represents the corresponding loss function.

4.3 Training procedure

After constructing the feature representations by the feature extractor, both the source and target domain aspects are passed into the domain discriminator. But, only the source instances are fed into the class discriminator. In our research, the aspects of the target domain also contain a sentiment polarity, but this information is not used in the training and remains unknown to the model up until the moment of testing. The performance of the DAT-LCR-Rot-hop++ is evaluated based on this testing accuracy. The benefit of being able to employ a model, which is trained only on the labels of a source domain, on a target domain gives the DANN approach an advantage over other methods.

The weights and biases are improved using the combined loss function, given by (11). This equation includes the $-\lambda$ multiplication in order to create sentiment discriminative and domain indiscriminate features. The domain discriminator uses ascending gradient to maximise this loss function, whereas the feature extractor and the class discriminator use descending gradient to minimise it. The exact training procedure is shown in Algorithm 1. The stopping condition is $\max(\text{acc}_{t-1}, \text{acc}_{t-2}) - \text{acc}_{t-3} > \epsilon$, which specifies that if the maximum of the accuracy of the previous epoch and the epoch before, minus the loss of three epochs ago is larger than ϵ continue with training. In other words, we continue if there is still a significant improvement.

The other hyperparameters besides λ in DAT-LCR-Rot-hop++ are the learning rates, η_k , the momentum terms, γ_k , the L2-regularisation terms, π_k , and the dropout rate. $k = d$ for the domain discriminator and $k = c$ for the feature extractor and class discriminator. First, η determines the rate at which the momentum optimiser converges. In addition, γ determines the influence of past gradient values on the current instance. Furthermore, π reduces overfitting. Fourth, as previously described, λ is a parameter that balances the trade-off between the discriminative objectives of the class and domain discriminator. Last, the dropout probability regulates the number of layer outputs to be randomly dropped from the network in order to prevent overfitting.

Because the dropout rate does not differ between the methods proposed in [33] and [34], this variable is kept at 0.3 in this research. The remaining hyperparameters (η_k , γ_k , π_k , and λ) are determined by a Tree-structured Parzen Estimator (TPE), which replaces the distribution of the initial observations with a non-parametric distribution by applying a threshold that splits the observations based on different densities [1].

As in the research performed in [33] and [34], the dimension of the word embeddings, $1 \times d$, is equal to 1×768 . For convenience, the number of nodes in the Bi-LSTMs bilinear and hierarchical attention layer are the same as in [33]. These are 300, 600, and 600, respectively. The number of hidden layers and cells in both the class and domain discriminator is optimised by TPE. The weights of the layers are initialised randomly using a normal distribution with a zero mean. The biases are set to zero at the start.

After the hyperparameters are initialised, DAT-LCR-Rot-hop++ is trained on the training set. The sentiment accuracy of the validation set is used to decide which combination of

Algorithm 1 Training procedure of Domain-Adversarial Learning.

$\epsilon = 0.50\%$

while stopping condition is not met **do do**

for each epoch **do do**

for each iteration i **do do**

- Sample approximately identical percentage batch of source domain, $S(x_i, y_i, d_i)$, and target, $T(x_i, y_i, d_i)$, data. n denotes the source domain batch size and N is the total batch size (source and domain combined).
- Feed input into feature extractor to obtain instance representations (r).
- Pass both $S(x_i, y_i, d_i)$ and $T(x_i, y_i, d_i)$ into domain discriminator and forecast the actual domain d_i . The predicted domain is defined as s_i . Afterwards update the parameters of the discriminator, θ_d , according to the loss function with ascending gradient:

$$\nabla_{\theta_d} [-\lambda(-\frac{1}{N} \sum_{i=1}^N d_i * \log(s_i) + \pi_d * \|\theta_d\|^2)]$$

- Last, feed $S(x_i, y_i, d_i)$ in class discriminator and predict the real label, y_i . The predicted sentiment is represented by p_i . Finally, adjust the parameters of both the feature extractor, θ_f , and sentiment classifier, θ_c , using the previously estimated domain discriminator parameters, $\hat{\theta}_d$, with descending gradient:

$$\nabla_{\theta_f, \theta_c} [-\lambda(-\frac{1}{N} \sum_{i=1}^N d_i * \log(s_i) + \pi_d * \|\theta_d\|^2)] +$$

$$-\frac{1}{n} \sum_{i=1}^n y_i * \log(p_i) + \pi_f * (\|\theta_f\|^2 + \|\theta_c\|^2)]$$

end for

end for

end while

hyperparameters achieves the best performance. We decided to let the program run 15 times for each source-target domain combination with different settings for the structure and the hyperparameters. Each run has maximum 50 epochs. The hyperparameter fine-tuning occurs twice. In the first step, λ is excluded, because we want to show the effect of λ on the cross-domain performance of the model. A higher λ should increase the domain-invariance of the features. As a result, λ will first be set to the value of 1.0 [8] in order to find the optimal values for the other parameters. After that the influence of λ is analysed and, then, all hyperparameters, including λ , are fine-tuned to define the best possible configuration. This setting is applied for the final training optimisation with a maximum of 200 epochs.

5 Evaluation

In Section 5.1, we first describe the influence of λ on the performance of DAT-LCR-Rot-hop++. Next, Section 5.2 provides a sensitivity analysis towards the structure of the class and domain discriminators. Then, in Section 5.3, the results for the final optimisation are shown. Last, Section 5.4 examines the performance of the algorithm when the neutral aspects are added to either the positive or negative class (to better balance classes).

5.1 Impact of λ

First, an estimate of the optimal settings for the algorithm is computed. These initial hyperparameters are shown in Table 2. The values are similar for every source-target domain combination. Similar settings are used to increase the comparability between the combinations. DAT-LCR-Rot-hop++ is run for seven incrementing values of λ with these settings, starting from 0.5 up until 1.1 with a step of 0.1 for each domain combination (Figure 2). The impact of the balance hyperparameter λ is visualised in the Figure 2. In these graphs, the dark blue (dark grey in black and white printing) line represents the labeling accuracies of the test set of the target domain, while the light orange (light grey in black and white printing) line shows the base performance when the majority group of the test sample was selected. Notice that the algorithm is only trained on the sentiments of the source domain, so one could argue that the base performance should equal the majority aspect class of the source domain. However, we use the more conservative view by defining the benchmark based on the target domain, which decreases the relative performance of our model. The model uses a maximum of 50 epochs.

When analysing Figure 2a and c, we observe that the classifying accuracy for the restaurant-laptop and laptop-restaurant domain combination is significantly higher than for the other four. Since the similarities between laptops and restaurants do not seem more prevalent than those between books and laptops, this might come across as a surprising result. However, both the laptop and restaurant domains are taken from the SemEval 2014 dataset [26] while the book domain is retrieved from the ALT 2019 [18]. First of all, these datasets share a common context and target text with words such as “service” and “quality”. Whereas the ALT 2019 dataset contains these target words 6 and 0 times, respectively, the words occur 59 and 85 times in the laptop set and 420 and 85 times in the restaurant domain, respectively. In addition, the (digital) language might have changed throughout these 5 years. Last, the fraction of neutral aspects in the book data test set is significantly higher than the neutral percentage in the training sets of both the restaurant and laptop domains with a percentage of 63.1 as compared to 17.7 and 19.8, respectively. This causes emotional phrases, for example “awesome”, to appear 5 times in the book domain as compared to 30 and 16 times in the laptop and restaurant domain, respectively. On these grounds, it is expected that the predicting score of the book domain in combination with either laptop or restaurant will result in lower scores compared to laptop-restaurant or restaurant-laptop.

Furthermore, the accuracy of the book as a target domain is worse than applying the book as a source domain. Because each domain has a disproportionate training set, it causes the neurons of the model to start with predicting the sentiment that occurs most often, especially in the early iterations. So, for both the restaurant and laptop domains this results in predicting

Table 2 Hyperparameter values for structure sensitivity analysis

Hyperparam	Value
η_d	0.01
$\eta_{c,f}$	0.01
γ_d	0.85
$\gamma_{c,f}$	0.85
π	0.001
λ	1.0
structure	2400-600

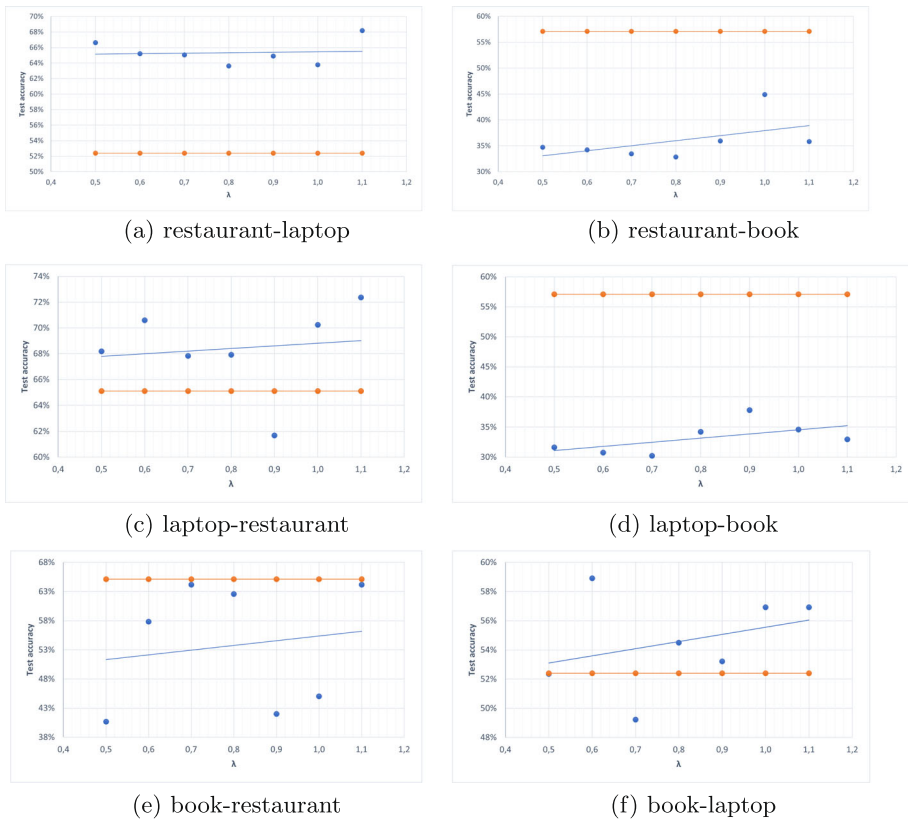


Figure 2 Labeling accuracies for different values of λ . Here the dark blue (dark grey in black and white printing) line is used for the target domain classification by our model and the light orange (light grey in black and white printing) line for the target domain classification by the majority classifier

a positive sentiment, which leads to a low score for the book target domain. On the other hand, using the book as a source domain does lead to acceptable performance. Not surprisingly, most neutral aspects are correctly classified in the book-restaurant and book-laptop combinations, with an average accuracy of 65% and 81%, respectively. Next, DAT-LCR-Rot-hop++ focuses on the second largest polarity percentage, the positive sentiments, which is the majority in both the restaurant and laptop domains. The drawback of this is the bad score for the negative polarities with an average accuracy of 3.2% and 0%, respectively. Besides this result, the restaurant-laptop combination does beat the base performance line at 52% with an average of 65%. The same applies to the laptop-restaurant case for which the observation at $\lambda = 0.9$ appears to be an outlier.

When looking at Figure 2a, we observe a scattered graph with an almost flat regression line. Accordingly, the coefficient of the OLS slope is 0.0061, which means that increasing λ with 1.0 increases the testing accuracy by 0.61%. The same holds for Figure 2c, which has a slope of 0.0204. One reason for this could be the previously mentioned overlap between the restaurant and laptop domain, thereby decreasing the difficulty of the cross-domain task and hence, making λ less important. In contrast to the flat regression lines of the restaurant-laptop and laptop-restaurant cases, the restaurant-book case depicts a clear positive linear trend with

a slope of 0.0973 in Figure 2b. The same applies to the graph in Figure 2d, which has a slope of 0.0691, showing the effect of λ .

As previously stated, employing the book as the target domain produces a poor labeling accuracy. Especially, the positive outlier in Figure 2b is an observation that proves the effect of the disproportionate data sets. Only during this run, DAT-LCR-Rot-hop++ was capable of predicting neutral sentiments correctly, which results in a neutral accuracy of 26% as compared to a maximum of 5% for the other runs with the book as a target domain. The competence to classify the neutral aspects precisely immediately leads to a significantly better performance with an accuracy of 45%, as compared to approximately 35%. The disproportionate sets cause DAT-LCR-Rot-hop++ to overfit on the training set and focus only on the major two polarities, resulting in low scores for the restaurant-book and laptop-book combination. Whereas there is a clear ascending performance for the training set, reaching percentages up until 92%, the maximum accuracy of the target domain is reached after approximately 100 epochs. The statistic then moves around this number with some large outliers in both directions, but never really improving. After some time, the accuracy starts to drop. The model becomes too much specified towards the information of the training set.

Both book-restaurant and book-laptop in Figure 2e and f, provide an ascending line with a slope coefficient of 0.0704 and 0.0488, respectively. However, the data points in Figure 2e are scattered, causing a standard deviation of 10%. Therefore, this positive relationship might be questioned in this case.

5.2 Sensitivity to discriminator structures

To estimate the sensitivity of the model to the neural structure of the domain and class discriminator, we examine the performance of three different structures. We run the DAT-LCR-Rot-hop++ for 125 epochs with the same hyperparameter settings as for the λ -analysis, given by Table 2. The output layer consists of the three sentiment classes. The three structures all have an input layer of 2400 nodes, but the hidden layers differ. The analysed set-ups are 1) No hidden layer, 2) one hidden layer with 600 neurons, 3) two hidden layers, one with 1200 nodes and the other consisting of 600 neurons.

The results for the examination are shown in Table 3. As one can see, the 2400-3 structure is not performing well for the classification problem. Due to the simplicity of the model, the DAT-LCR-Rot-hop++ is mostly predicting the majority class of the training domain. It is not trained enough to have good knowledge of other polarities as well. This leads to extremely low and high accuracies for specific classes. For instance, the restaurant-laptop 99% positive score and the book-laptop 0% negative accuracy. For five out of six domains, the dominant sentiment is predicted for at least 93% correctly, and the minor polarity scores for a maximum of 3%. Only once, for the laptop-book combination, there is some division over positive and negative aspects.

On the other hand, the 2400-600-3 system shows better scores. The accuracies are now more equally divided over the three different sentiments. Especially, the laptop-restaurant, book-restaurant, and book-laptop improvement within the division of correctly classified aspects is substantial. These three domain combinations lose some of their majority polarity accuracies but gain a significant better performance for the less prominent sentiments. The laptop-restaurant model sees an increase in accuracy of 39 percentage points for the negative aspects. So with an additional hidden layer, the algorithm is able to learn about the less dominant classes in the training sample. The performance of the 2400-1200-600-3 structure is somewhat similar to the 2400-600-3 model with some small score differences within the

Table 3 Test accuracies for DAT-LCR-Rot-hop++ model for three different structures of the class and domain discriminator

		test	train	max	base	pos	neu	neg
2400-3	rest-lapt	53%	62%	55%	52%	99%	1%	5%
	rest-book	34%	66%	35%	57%	99%	3%	6%
	lapt-rest	64%	48%	64%	65%	93%	2%	17%
	lapt-book	32%	50%	32%	57%	71%	5%	59%
	book-rest	19%	48%	19%	65%	2%	99%	0%
	book-lapt	33%	63%	35%	52%	15%	97%	0%
2400-600-3	rest-lapt	56%	63%	56%	52%	97%	3%	19%
	rest-book	32%	61%	33%	57%	97%	0%	7%
	lapt-rest	65%	54%	65%	65%	80%	16%	56%
	lapt-book	34%	53%	39%	57%	69%	9%	64%
	book-rest	33%	68%	39%	65%	25%	91%	4%
	book-lapt	43%	68%	43%	52%	38%	89%	0%
2400-1200-600-3	rest-lapt	53%	62%	54%	50%	92%	2%	22%
	rest-book	33%	64%	35%	55%	94%	2%	9%
	lapt-rest	62%	53%	63%	62%	74%	27%	50%
	lapt-book	30%	53%	32%	57%	70%	2%	60%
	book-rest	31%	55%	33%	65%	26%	93%	0%
	book-lapt	41%	65%	41%	52%	32%	91%	0%

“base” is the majority classifier in the target domain, test set, “pos”, “neu”, and “neg” are sentiment labels

sentiment class distribution. However, the increase of 4 minutes per iteration (50% extra) has to be considered, making the 2400-600-3 model a better fit for our classification task.

5.3 Final optimisation

From the results in Table 3, we decided to continue with a similar domain and class discriminator for all domain combinations. This structure consists of an input layer with 2400 neurons, one hidden layer with 600 nodes, and an output layer with 3 neurons, representing the sentiment classes. The final values of the hyperparameters for the final optimisation with maximum 200 iterations are defined in Table 4. Each domain combination is tested for the final prediction using these parameter settings. The results are shown in Table 5. As expected, the accuracies improve for each source and target domain model as compared to the previous run with maximum 50 epochs and optimal hyperparameters including λ . The training label accuracy increases from 84% up to 90% for the book-laptop domain. In addition, the maximum testing accuracy of 78% for the restaurant-laptop case is a 7 percentage points improvement from the previous 71%. The ratios of correctly predicted polarities follow the previously seen distribution.

The performance of the restaurant-laptop domain increased significantly, which results in a total test accuracy of 75%. The relevance of not requiring any labeled target data should not be underestimated when comparing it with other research because this ability reduces labeling costs significantly. Specifically, the outcomes for the book-restaurant case are promising. Both domains are not closely related in terms of sentiment distribution, but the model achieves

Table 4 Hyperparameter values for final model estimation

Hyperparam	re-la	re-bo	la-re	la-bo	bo-re	bo-la
η_d	0.01	0.03	0.03	0.01	0.01	0.03
$\eta_{c,f}$	0.005	0.005	0.03	0.01	0.005	0.03
γ_d	0.90	0.85	0.80	0.90	0.85	0.90
$\gamma_{c,f}$	0.90	0.90	0.85	0.85	0.85	0.85
π	0.001	0.001	0.001	0.001	0.001	0.0001
λ	1.1	0.8	1.1	0.6	1.1	0.6
structure	2400-600	2400-600	2400-600	2400-600	2400-600	2400-600

“re”, “la”, and “bo” represent restaurant, laptop, and book domain, respectively

an encouraging test accuracy of 69%, which is an improvement of 5 percentage points as compared to the value after the previous runs. Interestingly, the fraction of correctly labeled sentiments are more balanced, instead of one polarity that is driving the results.

5.4 Extension on neutral sentiments

Looking at Table 5, one can conclude that correctly classifying neutral sentiments is difficult for the algorithm. One reason for this might be the disproportionate datasets, especially in terms of neutral aspects. Another problem is the difference in proportions between source and target domains. Third, one can argue that predicting neutral sentiments is more difficult than labeling negative or positive polarities. For example, “amazing”, and “terrible” are clearly positive and negative words, respectively, while “fine” and “okay” can indicate both positive and neutral words. As a result, we also examine the model with the standard binary classification of either positive or negative. To do this, we analyze three cases: 1) the “Base” case with the neutral sentiment included, 2) the neutral aspects added to the positive polarity, and 3) the neutral and negative polarities combined. Each specification is tested using the same settings as for the structure analysis in Table 2. The algorithm is run for 125 epochs.

Table 6 shows the new polarity proportions. First of all, we hypothesise that the scores will be better for the binary case because there are two options to choose from instead of one. Furthermore, from this table, one can expect that the “neutral to positive” adaption produces better results compared to “neutral to negative”, due to the dominant component of positive polarities. Last, one can assume that the book domain performs significantly better in terms of source and target domain as the differences between the distributions of the dataset and the restaurant and laptop domains are smaller.

Table 5 Test accuracies for DAT-LCR-Rot-hop++ model

	test	train	max	base	pos	neu	neg
rest-lapt	75%	85%	78%	52%	92%	36%	81%
rest-book	35%	86%	46%	57%	69%	7%	86%
lapt-rest	70%	80%	74%	65%	86%	5%	81%
lapt-book	39%	88%	49%	57%	83%	8%	69%
book-rest	69%	86%	74%	65%	75%	78%	37%
book-lapt	58%	90%	64%	52%	73%	79%	0%

“base” is the majority classifier in the target domain test set, “pos”, “neu”, and “neg” are sentiment labels

Table 6 The distribution of the aspect sentiment polarities of the three domains for the neutral extension

		Restaurant		Laptop		Book	
		Train	Test	Train	Test	Train	Test
Base	Positive	60.1%	65.1%	42.7%	52.4%	25.8%	32.3%
	Neutral	17.7%	17.5%	19.8%	26.2%	63.1%	57.1%
	Negative	22.2%	17.4%	37.5%	21.4%	11.1%	10.6%
Neutral to positive	Positive	77.8%	82.6%	62.5%	78.6%	88.9%	89.4%
	Negative	22.2%	17.4%	37.5%	21.4%	11.1%	10.6%
Neutral to negative	Positive	60.1%	65.1%	42.7%	52.4%	25.8%	32.3%
	Negative	39.9%	34.9%	57.3%	47.6%	74.2%	67.7%
	Total	3600	1122	2250	701	2700	804

The results of the neutral extension are given by Table 7. The percentage scores for the base case are significantly lower than in Table 5. This can be the consequence of a mix of reasons. First, the optimization is run for 125 epochs instead of 200. Second, the optimal hyperparameters for each individual domain combination are not used, but a general setting to increase comparability instead. Third, it can be bad luck. The algorithm could end up in a local minimum without further exploring other minima.

Table 7 Test accuracies for DAT-LCR-Rot-hop++ model for the neutral extension

		test	train	max	base	pos	neu	neg
Base	rest-lapt	56%	65%	56%	52%	95%	2%	26%
	rest-book	34%	66%	35%	57%	99%	3%	6%
	lapt-rest	61%	59%	63%	65%	68%	30%	67%
	lapt-book	34%	55%	34%	57%	79%	6%	42%
	book-rest	35%	71%	39%	65%	27%	94%	6%
	book-lapt	43%	69%	44%	52%	39%	86%	1%
Neutral to positive	rest-lapt	80%	83%	80%	79%	98%	NA	13%
	rest-book	85%	82%	89%	89%	95%	NA	6%
	lapt-rest	82%	72%	83%	83%	91%	NA	39%
	lapt-book	75%	71%	83%	89%	81%	NA	27%
	book-rest	83%	90%	83%	83%	100%	NA	2%
	book-lapt	81%	69%	83%	79%	98%	NA	16%
Neutral to negative	rest-lapt	63%	71%	63%	52%	90%	NA	33%
	rest-book	54%	70%	57%	68%	88%	NA	38%
	lapt-rest	65%	73%	67%	65%	57%	NA	80%
	lapt-book	70%	71%	71%	68%	55%	NA	77%
	book-rest	44%	82%	53%	65%	16%	NA	96%
	book-lapt	62%	81%	62%	52%	35%	NA	91%

“base” is the majority classifier in the target domain, test set, “pos”, “neu”, and “neg” are sentiment labels “Base” is the normal case with neutral aspects. “Neutral to positive” represents the binary problem with neutral sentiments added to positive. Vice versa for “Neutral to negative”

As expected, the test scores are higher for every run of both the “neutral to positive” and “neutral to negative”. The average difference in percentage between the base case is 37% and 16%, respectively. The same holds for the train classification performance. Transforming the three-dimensional polarity labeling issue to a binary classification problem improves the accuracy substantially.

Furthermore, it is not surprising to see that the positive transformation performs better than the negative transformation for each of the domain combinations. The extreme towards positive polarity distributed datasets of all the domains result in the algorithm predicting almost all positive aspects correctly. A 100% score is reached for the book-restaurant and 98% for the book-laptop combination. Both are accompanied by a low score for the negative sentiment. In addition, interesting to see that the positive opinions are classified as the least accurate for the laptop source domain because its train dataset is the most equally divided. Also, one can notice the minimum 27% accuracy of negative aspects for the laptop source domain compared to the others. This again proves the importance of similarity between sentiment distribution between source and target domains.

Last, Table 7 shows indeed that the score for the book domain as either the source or target domain is significantly higher with the neutral extension than for the base case. Interesting to observe that the accurate labeling of a polarity depends on the amount of a specific polarity in the training set to be able to predict this label category correctly in the test set. The book-restaurant combination for the positive adaption returns a 100% and 2% accuracy for the positive and negative aspects, respectively. However, the book-restaurant combination for the negative transformation results in a score of 16% and 96%, respectively. It shows that the DAT-LCR-Rot-hop++ requires a substantial level of a certain sentiment class to have the knowledge to label the corresponding aspects correctly.

6 Conclusion

The important role of user-generated content on the Web increases the relevance of ABSA, and in particular ABSC. Since obtaining labelled target data is extremely costly, new models should be developed that can be employed in a variety of domains. The state-of-the-art LCR-Rot-hop++ structure forms the basis of our proposed DAT-LCR-Rot-hop++, which adds an adversarial component based on DANN. Based on our results, we show that the domain invariance implemented through DAT-LCR-Rot-hop++, in general, can improve performance over the target data, especially for similar domains.

All accuracy scores for the restaurant-laptop, laptop-restaurant, and book-restaurant domains exceed 70%. So in half of the considered source-target domain cases, DAT-LCR-Rot-hop++ is able to classify polarities properly, but it depends on which combination of domains is used. Domains with similar polarity distributions seem to benefit the most from the proposed approach.

To further examine our method, we investigate the effect of adding the neutral aspects to either the positive or negative side, creating a binary classification problem. Classifying neutral aspects appears to be a harsh task for the neural network, looking at the results, as this class is poorly represented in two out of our three domains. The transformation to the binary problem improves the results as expected. In addition, it again shows that similarity between the source and target domain is crucial.

Author Contributions Joris Knoester did the implementation. Flavius Flasincar and Maria Mihaela Trusca did review the manuscript. All authors wrote the manuscript.

Funding Not applicable.

Availability of data and materials The code and the dataset are available at <https://github.com/jorisknoester/DAT-LCR-Rot-hop-PLUS-PLUS>.

Declarations

Ethical Approval Not applicable.

Competing interests The authors declare no competing interests.

References

- Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: 25th Annual conference on neural information processing systems (NIPS 2011), pp. 2546–2554. Curran Associates (2011)
- Brauwiers, G., Frasinca, F.: A survey on aspect-based sentiment classification. *ACM Comput. Surv.* **55**(4), 65:1–65:37 (2023)
- Chapman, B.: Gamestop: reddit users claim victory as \$13bn hedge fund closes position, accepting huge losses (2021). <https://www.independent.co.uk/news/business/gamestop-share-price-reddit-hedge-fund-melvin-capital-b1793543.html>
- Chen, Y., Tong, Z., Zheng, Y., Samuelson, H., Norford, L.: Transfer learning with deep neural networks for model predictive control of HVAC and natural ventilation in smart buildings. *J. Clean. Prod.* **254**, 119866 (2020)
- Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: 2012 IEEE Conference on computer vision and pattern recognition (CVPR 2012), vol. 1, pp. 3642–3649. IEEE Computer Society (2012)
- Devlin, J., Chang, K., Lee, K., Huang, D., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: 2019 Conference of the North American chapter of the association for computational linguistics (NAACL-HLT 2019), pp. 4171–4186. ACL (2019)
- Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: 32nd International conference on machine learning (ICML 2015), vol. 37, pp. 1180–1189. PMLR (2015)
- Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **17**, 59:10–59:35 (2016)
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. In: 28th Annual conference on neural information processing systems (NIPS 2014), pp. 2672–2680 (2014)
- Hendricks, D.: Complete history of social media: then and now (2013). <https://smallbiztrends.com/2013/05/the-complete-history-of-social-media-infographic.html>
- Hong, W., Wang, Z., Yang, M., Yuan, J.: Conditional generative adversarial network for structured domain adaptation. In: 8th IEEE International conference on computer vision and pattern recognition (CVPR 2018), vol. 12, pp. 1335–1344. IEEE (2018)
- Hong, Y., Zhou, W., Zhang, J., Zhu, Q., Zhou, G.: Self-regulation: employing a generative adversarial network to improve event detection. In: 56th Annual meeting of the association for computational linguistics (ACL 2018), pp. 515–526. ACL (2018)
- Johansson, F., Shalit, U., Sontag, D.: Learning representations for counterfactual inference. In: 33rd International Conference on Machine Learning (ICML 2016). JMLR Workshop and conference proceedings, vol. 48, pp. 3020–3029. JMLR (2016)
- Kamath, S., Gupta, S., Carvalho, V.: Reversing gradients in adversarial domain adaptation for question deduplication and textual entailment tasks. In: 57th Annual meeting of the association for computational linguistics (ACL 2019), pp. 5545–5550. ACL (2019)

15. Knoester, J., Frasinca, F., Trusca, M.M.: Domain adversarial training for aspect-based sentiment analysis. In: 23rd International conference on Web information systems engineering (WISE 2022). LNCS, vol. 13724, pp. 21–37. Springer (2022)
16. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: 29th Conference on artificial intelligence (AAAI 2015), vol. 29, pp. 2267–2273. AAAI Press (2015)
17. Liu, C., Belkin, M.: Accelerating SGD with momentum for over-parameterized learning. In: 8th International conference on learning representations (ICLR, 2020). OpenReview.net (2020)
18. Álvarez López, T., Fernández-Gavilanes, M., Costa-Montenegro, E., Bellot, P.: A proposal for book oriented aspect based sentiment analysis: comparison over domains. In: 23rd International conference on applications of natural language to information systems (NLDB 2018). LNCS, vol. 10859, pp. 3–14. Springer (2018)
19. Maat, E.D., Krabben, K., Winkels, R.: Machine learning versus knowledge based classification of legal texts. In: 23rd Annual conference on legal knowledge and information systems (JURIX 2010), vol. 223, pp. 87–96. IOS Press (2010)
20. Mauro, M., Mazzia, V., Khalil, A., Chiaberge, M.: Domain-adversarial training of self-attention based networks for land cover classification using multi-temporal sentinel-2 satellite imagery. *Remote Sens.* **13**(13), 2564 (2021)
21. Pan, S., Yang, Q.: A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **22**, 1345–1359 (2009)
22. Pang, B., Lee, L.: A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: 42nd Annual meeting of the association for computational linguistics (ACL 2004), pp. 271–278. ACL (2004)
23. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: 2002 Conference on empirical methods in natural language processing 2002 (EMNLP 2002), pp. 79–86. ACL (2002)
24. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: Semeval-2015 task 12: aspect based sentiment analysis. In: 9th International workshop on semantic evaluation (SemEval 2015), pp. 486–495. ACL (2015)
25. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Zafra, S., Eryigit, G.: Semeval-2016 task 5: aspect based sentiment analysis. In: 10th International workshop on semantic evaluation (SemEval 2016), pp. 19–30. ACL (2016)
26. Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: Semeval-2014 task 4: aspect based sentiment analysis. In: 8th International workshop on semantic evaluation (SemEval 2014), pp. 27–35. ACL (2014)
27. Schouten, K., Frasinca, F.: Survey on aspect-level sentiment analysis. *IEEE Trans. Knowl. Data Eng.* **28**, 813–880 (2016)
28. Schouten, K., Frasinca, F.: Ontology-driven sentiment analysis of product and service aspects. In: 15th International conference of European semantic Web (ESWC 2018). LNCS, vol. 10843, pp. 608–623. Springer (2018)
29. Taboada, M., Brooke, J., Tofloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. *Comput. Linguist.* **37**, 267–307 (2011)
30. Tankovska, H.: Social media - statistics and facts (2021). <https://www.statista.com/topics/1164/social-networks/>
31. Thet, T., Na, J., Khoo, C.: Aspect-based sentiment analysis of movie reviews on discussion boards. *J. Inf. Sci.* **36**, 823–848 (2010)
32. Towell, G., Shavlik, J.: Knowledge-based artificial neural networks. *Artificial. Intelligence* **70**, 119–165 (1994)
33. Trusca, M., Wassenberg, D., Frasinca, F., Dekker, R.: A hybrid approach for aspect-based sentiment analysis using deep contextual word embeddings and hierarchical attention. In: 20th International conference on Web engineering (ICWE 2020). LNCS, vol. 12128, pp. 365–380. Springer (2020)
34. Wallaart, O., Frasinca, F.: A hybrid approach for aspect-based sentiment analysis using a lexicalized domain ontology and attentional neural models. In: 16th International conference of European semantic Web (ESWC 2019). LNCS, vol. 11503, pp. 363–378. Springer (2019)
35. Wang, F., Zhang, Q.: Knowledge-based neural models for microwave design. *IEEE Transactions on Microwaves Theory and Techniques* **45**, 2333–2343 (1997)
36. Wang, Z., Huang, M., Zhao, L., Zhu, X.: Attention-based LSTM for aspect-level sentiment classification. In: 2016 Conference on empirical methods in natural language processing (EMNLP 2016), pp. 606–615. ACL (2016)

37. Wu, Y., Inkpen, D., El-Roby, A.: Co-Regularized Adversarial Learning for Multi-Domain Text Classification. In: 2022 International conference on artificial intelligence and statistics (AISTATS 2022), pp. 6690–6701. PMLR (2022)
38. Yanase, T., Yanai, K., Sato, M., Miyoshi, T., Niwa, Y.: bunji at SemEval-2016 task 5: neural and syntactic models of entity-attribute relationship for aspect-based sentiment analysis. In: 10th International workshop on semantic evaluation (SemEval 2016), pp. 289–295. ACL (2016)
39. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: 27th Annual conference on neural information processing systems (NIPS 2014), vol. 27, pp. 3320–3328. Curran Associates (2014)
40. Yuan, J., Zhao, Y., Qin, B., Liu, T.: Learning to share by masking the non-shared for multi-domain sentiment classification. *Int. J. Mach. Learn. Cybern.* **13**(9), 2711–2724 (2021)
41. Zhang, W., Ouyang, W., Li, W., Xu, D.: Collaborative and adversarial network for unsupervised domain adaption. In: 2018 Conference on computer vision and pattern recognition (CVPR 2018), pp. 3801–3809. IEEE (2018)
42. Zhang, Y., Qiu, Z., Yao, T., Liu, D., Mei, T.: Fully convolutional adaption networks for semantic segmentation. In: 2018 International conference on computer vision and pattern recognition (CVPR 2018), pp. 6810–6818. IEEE (2018)
43. Zheng, L., Zhang, Y., Wu, Y., Wei, Y., Yang, Q.: End-to-end adversarial memory network for cross-domain sentiment classification. In: 26th International joint conference on artificial intelligence (IJCAI 2017), pp. 2237–2243. IJCAI (2017)
44. Zheng, S., Xia, R.: Left-center-right separated neural network for aspect-based sentiment analysis with rotatory attention. arXiv preprint [arXiv:1802.00892](https://arxiv.org/abs/1802.00892) (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.