Domain Adversarial Training for Aspect-Based Sentiment Analysis

ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics
Bachelor Thesis BSC^2

Name student: Joris Knoester

Student ID number: 479288

Supervisor: Flavius Frasincar

Second assessor: Mehmet Hakan Akyüz

Date final version: July 4, 2021

Abstract

The continuously expanding digital possibilities, increasing number of social media platforms, and growing interest of companies in online marketing, increase the popularity of Aspect-Based Sentiment Analysis (ABSA). ABSA focuses on predicting the sentiment of an aspect in a text. There are multiple relevant applications of ABSA. However, each task requires training the model on a new domain. In the perfect world we would have labeled data for every existing domain. But, acquiring annotated training data is extremely costly. Transfer learning resolves this issue by building models that can be employed on different domains. This paper improves the state-of-the-art LCR-Rot-hop++ model for ABSA introduced by Trusca et al. (2020) with the methodology of Domain Adversarial Training (DAT) as proposed by Ganin et al. (2016) in order to create a deep learning adaptable cross-domain structure, called the DAT-LCR-Rot-hop++. The major advantage of the DAT-LCR-Rot-hop++ compared to other modern models is the fact that it does not require any labeled target data during training. The results are obtained for six different domain combinations with testing accuracies ranging from 37% up until 77%, showing both the limitations and benefits of this approach. Once the DAT is able to find the similarities between both domains it produces good results, but if the domains are too far off, it is not capable of generating domain-invariant features.

Contents

Chapter 1

Introduction

This chapter describes the topic and relevance of this research. Section 1.1 provides an introduction to ABSA and states the problem. Section 1.2 gives brief information about the the model of this thesis and specify its research objectives. Last, Section 1.3 covers the structure of chapters this paper.

1.1 Problem Statement

Social media has become an inevitable part of our lives. Since the introduction of the first social media platform, Six Degrees in 1997 (Hendricks, 2021), more and more people communicate in a digital manner. Last year, over 3.7 billion people, which accounts for 48% of the world population, has exchanged information on a social media platform (Tankovska, 2020).

Evidently, brand image is of great importance for business firms. Due to the continuously growing society of online users, more companies are getting aware of the essential role that social media plays in this. The increase in users has also expanded the amount of opinionated messages. The online buyers express their ideas and feelings more openly than ever before via survey responses, online reviews, and social media conversations. This creates the opportunities for companies to understand the consumer's wishes and adjusts the products to their needs. On the other hand it can help potential customers make better decisions when buying products.

When analysing product reviews, a company is interested in the customer's opinions about the complete product, but also the client's feelings towards specific features. This is a job for Aspect-Based Sentiment Analysis (ABSA) (Thet et al., 2010). The major tasks of ABSA are target extraction (TE), aspect detection (AD), and target sentiment classification (SC) (Schouten & Frasincar, 2016). The TE task is dealing with the selection of targets, AD is concerned with identifying aspects that refer to target's sentiments, and SC focuses on classifying the user's sentiment with respect to the aspects. This paper concentrates on the last task, SC, and is called Aspect-Based Sentiment Classification (ABSC).

Whereas sentiment analysis (SA) focuses on the overall opinion of a review (Liu, 2020), ABSC concentrates on a user's sentiment towards an individual aspect. Take for instance the sentence "the atmosphere and service was terrible, but the food was fine". The overall polarity should be classified as negative, but

the opinion about the food differs. In this case, the "atmosphere", "service", and "food" are all *target words* and "terrible" and "fine" are the expressions that give *context* to these target words. As you can see, 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, increasing their customer services in an efficient manner. 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 (Chapman, 2021). Lastly, knowing the opinion of previous customers can help potential clients make better informed decisions for buying certain products.

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 (Pan & Yang, 2009). This approach concentrates on training a model on a related source domain and then predicting the sentiments for a different target domain. So, for example training a neural network on labeled movies reviews data and afterwards adapting and testing this trained model on book reviews. In this case, both domains will probably contain target words such as "protagonist" and "dialogue" and polarity words being "good-looking" and "thrilling". By memorising these shared features, the network is able to apply the gained knowledge from the source domain on the new target domain.

1.2 Research Objectives

Several state-of-the-art cross-domain models rely on Domain-Adversarial Neural Networks (DANN) introduced by Ganin et al. (2016). These neural networks are applied on diverse tasks, ranging from textual entailment analysis (Kamath et al., 2019) to image classification (Zhang et al., 2018). However, little research is available on ABSA using Domain-Adversarial Training (DAT). To our knowledge, there is no current model that trains domain-invariant features for Aspect-Based Sentiment text Classification based on DANN.

In terms of sentiment classification, the polarities of the aspects in the reviews can be classified as either positive, neutral, or negative. Prior methods apply a Support Vector Machine (Pang & Lee, 2004) in order to predict these emotions, but due to its shortcomings this system was replaced by knowledge-based models (Taboada et al., 2011) and deep learning algorithms (Lai et al., 2015). Whereas the deep learning methods are flexible, knowledge-based models require more manual labour, but achieve better results (de Maat et al., 2010). Since a combination of both approaches benefits from the advantages of these solutions, several researchers merge the methods into a hybrid model (Towell & Shavlik, 1994) (Wang & Zhang, 1997). The methodology of this paper is based on the hybrid HAABSA++ model proposed by

Trusca et al. (2020), a state-of-the-art approach that produces excellent results for the commonly used SemEval 2015 (Pontiki et al., 2015) and 2016 (Pontiki et al., 2016) 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, which is focused on restaurants, is eliminated, leaving the LCR-Rot-hop++ model.

In addition, this research applies the structure of DAT as proposed by (Ganin et al., 2016). More specifically, the neural network is trained concurrently on labeled instances of a source domain and unlabeled aspects from a target domain. These instances are the input for the bi-LSTM layers and the subsequent attention mechanisms. After the hierarchical attention layer, a Gradient Reversal Layer (GRL) and both a class and domain discriminator are incorporated. The GRL acts as adversary by increasing the difficulty of predicting the text's domain, which is the task of the domain discriminator. It adjusts the incoming loss gradient in exactly the opposite way as desired by the domain discriminator. It does this by multiplying the loss with a negative balance parameter λ, which leads to ineffective feature extraction layers for domain classification in the LCR-Rot-hop++. This unites the feature maps of the source and target domain. As a result, the optimal representation layers produce domain-invariant feature vectors. The other steps of the LCR-Rot-hop++ model remain unchanged, hereby referring to the contextual BERT word embeddings, the rotary attention mechanism, and the rotary hierarchical attention layer. We call the new established system Domain Adversarial Training LCR-Rot-hop++ abbreviated as DAT-LCR-Rot-hop++.

The main contribution of this research on current literature is the capability of predicting aspect-based sentiment classifications on target aspects without requiring annotated target data by employing a DANN. Following the general DANN approach, this paper uses both labeled source and unlabeled target domain aspects to obtain domain indiscriminative representations. As far as we know, there is no advanced neural network as LCR-Rot-hop++ for ABSA that exploits DANN in a cross-domain setting. Because of the excellent performance of the LCR-Rot-hop++ and proven solution of DAT, we expect that the proposed DAT-LCR-Rot-hop++ achieves a good classification accuracy on previously unseen domains.

Last, it is interesting to analyse the effect of different values for λ on the ability to classify target domain data. In general, the larger value for λ, the more domain-invariant the weights and biases of the features of LCR-Rot-hop++ are. But, at the same time it could lower the capability of the feature extractor to produce features, which are useful for sentiment classification.

1.3 Thesis Structure

All source code and data can be retrieved from https://github.com/jorisknoester/DAT-LCR-Rot-hop-PLUS-PLUS. The rest of this paper is structured in the following way. First, Chapter 2 gives the relevant literature concerned with ABSC, the different components of the LCR-Rot-hop++ model, and the theory behind transfer learning. Second, Chapter 3 provides a short description of the used data

together with a couple of descriptive statistics. Third, Chapter 4 concentrates on the methodology of this research for which the results are reported in Chapter 5. Last, Chapter 6 provides our conclusion and suggestions for future work.

Chapter 2

Related Work

This chapter covers the relevant literature that is written on ABSC, the components of the LCR-Rot-hop++ model, and transfer learning. Hence, this chapter is divided in three parts, each describing one of the three mentioned elements.

2.1 ABSC

ABSA, which includes ABSC, is a popular field of research (Schouten & Frasincar, 2016). It is concerned with classifying a person's sentiment towards specific aspects in a sentence. The polarities of texts can be classified as either positive, neutral, or negative, a process that is defined as sentiment classification. This is a Natural Language Processing (NLP) technique. Sentiment analysis was first introduced by Pang et al. (2002) and has been a hot topic ever since. Traditional methods focus on basic machine learning such as a Support Vector Machine (SVM) model (Pang & Lee, 2004). Disadvantages of this mechanism include unsuitability to large data sets, and no probalistic explanation for the classification (Karamizadeh et al., 2014).

On the other hand, machine learning algorithms tend to be more effective at performing classification tasks. The phrase "deep learning" was established by Dechter (1986). This concept was enriched by the back-propagation algorithm introduced by LeCun (1989). Due to the excellent performance, a lot of research has been completed on different training strategies (Larochelle et al., 2009) (Glorot & Bengio, 2010). One of the main advantages of deep learning is its ability to execute feature engineering on its own by its hidden layers. This makes it suitable for image (Ciresan et al., 2012) and text (Lai et al., 2015) classification methods.

Separate from the above, a third approach, which does not concern machine learning, is knowledge-based sentiment classification. This solution relies more on domain expertise and human intelligence and is therefore less flexible. Verhagen et al. (2012) perform a literature review on knowledge-based engineering models and point out its inability to reuse the knowledge-based model for other problems and the importance of knowledge loss. With respect to text classification, you can imagine that the definition and hence polarity of certain words, such as "bad" will not be forgotten. But if a knowledge-

based model is not continuously updated according to the changes in language, at a certain point it will start to fail. Updating the knowledge throughout time is a solution, but this requires extra manual work.

Nevertheless, De Maat et al. (2010) argue that knowledge-based systems provide higher prediction scores for domain-specific documents. In line with this conclusion, IJntema et al. (2010) develop a domain ontology to provide news recommendations to online users. Moreover, Schouten and Frasincar (2018) first classify the sentiments of the instances by using an ontology-based approach. This ontology is the starting point of the HAABSA++ model and, as explained previously, is not considered in this research. Yanase et al. (2016) observe the competition from a different perspective and state that both methods are in fact complementary. Consequetively, Schouten and Frasincar (2018) produce a hybrid which incorporates both a domain ontology and a deep learning neural network. After several additional improvements, this has resulted in the HAABSA++ method.

2.2 HAABSA++

HAABSA++ is developed by Trusca et al. (2020). It consists of several individual components, carefully designed to work well together. First of all, the input of the neural network are BERT contextual word embeddings (Devlin et al., 2019). Word embeddings are a crucial aspect of NLP. Maas et al. (2011) are one of the first researchers to develop a word feature map that is adequate for sentiment classification. Subsequently, a couple of years ago, there was a huge breakthrough in this field of investigation introduced by Devlin et al. (2019). They established the Bidirectional Encoder Representations from Transformers (BERT), which is now widely used in NLP (Rietzler et al., 2020) (Li et al., 2019). It is a Google trained model that uses the mathematics of encoder and decoders together with masking and can be applied to a broad variety of tasks, such as language inference and question answering. Trusca et al. (2020) improve the hybrid LCR-Rot-hop approach (Wallaart & Frasincar, 2019) by replacing the original non-contextual Glove word embeddings with deep contextual BERT word representations.

The word vectors are the input for a Left-Center-Right bidirectional Long Short-Term Memory (LSTM) model with rotary attention (LCR-Rot) introduced by Zheng and Xia (2018). This model is a specification of Reccurent Neural Networks (RNN). RNNs have been applied to a diversity of tasks, such as speech recognition (Mikolov et al., 2010), text processing (Ying et al., 2017), and video captioning (Zhao et al., 2019). RNNs consist of an input, hidden, and output layer and is capable of forecasting current output based on historical features. This exceeds the competence of standard feed-forward networks. The ability of RNN to maintain a memory is a consequence of the connections between its hidden layers.

Following from this, the LSTM model proposed by Hochreiter and Schmidhuber (1999), is one of the first important modifications of RNN. In this structure, the hidden layer cells are substituted by memory LSTM neurons. These LSTM cells provide better performance than standard RNN cells due to their capability of storing weights activation scores for a significant amount of time. Huang et al. (2015) adapt the LSTM neurons to become bi-LSTM-cells, which are able to efficiently use past and future features instead of only historical ones.

In the LCR-Rot-hop++ method, the bi-LSTM cells are accompanied by an attention mechanism, proposed by Wang et al. (2016). This attention layer is able to concentrate and put focus on specific parts of a sentence, thereby limiting the influence of the less important words, such as "chair" as compared to "food" in a restaurant. On top of this attention layer, a hierarchical attention layer is employed, enabling the model to process the text on sentence-level, bringing together the local sentence representations (Yang et al., 2016). The last component of the LCR-Rot-hop++ model is its rotary system along the two attention layers. This increases the interaction between targets and contexts by sharing information in order to capture the most indicative sentiment words.

2.3 Transfer learning

Transfer learning is an important machine learning technique (Pan & Yang, 2009). In order to avoid ambiguity, domain adaption or cross-domain processing is used for transfer learning in NLP. It focuses on storing information from one data set and applying this knowledge on another. Several researchers in multiple scientific areas try to utilise this method, ranging from biology (Ribeiro et al., 2020) and energy prediction (Fang et al., 2021) to sentiment classification (Yuan et al., 2021). Since obtaining classified data is costly and time-consuming, it is crucial that new models will be developed that are trained on a related labeled source domain and then capable of providing reliable results for other domains. In this case, you only require one classified dataset to train the model and the resulting neural network can then be used to predict outcomes for other unlabeled 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 (Chen et al., 2020). This approach is based on the fact that the higher layer neurons tend to specialize more towards the target domain, while the lower hidden layers generate more common word features (Yosinki et al., 2014). In addition, others use pre-trained models for feature extraction and pre-trained models for weight initialisation (Fan et al., 2020). Both concepts depend on the fact that the source domain contains valuable universal information. Hence, multiple weights and biases can be fixed after training, only a few should be adjusted for the target domain, thereby minimising the required amount of needed labeled target data.

Furthermore, Tian et al. (2021) construct an end-to-end cross-domain ABSA model by implementing a random parameter generator for the bi-LSTM layers and two CRF layers. The generator generates cross-domain parameters by using domain representations that can learn domain-dependent information via training both domains. These parameters are then applied to reform the bi-LSTM layer. Next, the CRFs are each specialised in predicting the class of either the source or target domain.

A state-of-the-art method, the BertMasker, introduced by Yuan et al. (2021) uses the mathematics of masking as proposed by Devlin et al. (2019) for their BERT Base network. The idea behind this structure is that the BertMasker is able to mask domain-related words. This transforms the remaining sentence text to be domain-invariant, but at the same time still maintains its most sentiment-explicit words. In terms of performance, the authors achieve positive results, beating most of the currently available models.

On the other hand, other scientists developed a domain adapting network by creating counterfactual

features (Johansson et al., 2016). These counterfactual depictions reduce the inductive bias of the source domain. First, a labeled dataset of the source domain is used as input to add domain-aware features. Next, a discriminator is trained to recognize both the source and target domain. Last, by editing the gradient, counterfactual representations are created. You can imagine that if you plot the positive instances of the source domain and target domain in a multidimensional graph, both positive groups differ in terms of position and coordinates. The same applies to the negative observations. Constructing a line to divide the positive and negative aspects of both domains in this map would result in a crooked border. The designed positive counterfactuals bridge the dimensional gap between the positive classified instances of the source and target domain. At the same time, the negative counterfactuals connect the negative aspects of both domains. This procedure alters and improves the sentiment frontier, which can now be drawn as a straight line that separates the aspects in the multi-dimensional feature map.

Differently from the previous works, in this research we use the methodology of Generative Adversarial Networks (GAN). This last solution is introduced by Goodfellow et al. (2014) and has shown superior performance in a broad range of scientific areas. Whereas, Zhang et al. (2019) propose a self-attention GAN to classify images, Hong et al. (2018) introduce a system that predicts the event that a document is referring to. In addition, Zheng et al. (2017) develop an Adversarial Memory Network (AMN) for textual cross-domain sentiment classification, outperforming other existing approaches. Ganin et al. (2015) used the logic of GAN and developed DANN. A DANN 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 percentages and batch size (Ganin et al., 2015). The advantage that you do not need annotated target data makes DANN very valuable for future cross-domain deep learning problems. DANN is therefore an important contribution to the existing machine learning techniques.

Chapter 3

Data

In this paper, two different data sets are used. These are the Semantic Evaluation (SemEval) 2014 (Pontiki et al., 2014), and the Amazon/LibraryThing (ALT) 2019 (Álvarez-López et al., 2018). The SemEval 2014 includes information about the restaurant and laptop domain, and the ALT contains the data for the book domain. The SemEval datasets are widely used for NLP tasks (Zhao et al., 2014) (Trusca et al., 2020), thereby increasing the comparability of this paper to other researches. We did not choose SemEval 2015 (Pontiki et al., 2015) and SemEval 2016 (Pontiki et al., 2016) data sets as these do not define targets for computing sentiments for all the individual domains, while our employed neural network requires these targets to be present. The used data sets will be described in more detail in Section 3.1-3.3.

The partitioning of the data into a training and test set is taken from van Berkum et al. (2021). 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 is shown for each domain in Appendix A.

DAT-LCR-Rot-hop++ requires aspects from two domains to be passed through the model. These domains are defined as the source domain and the target domain. The source domain contains labeled instances, such that each observation has a sentiment attached to it. On the other hand, the target domain is unlabeled, so the model does not know the polarity of these aspects. The neural network does perceive whether or not the instance comes from the source domain or the target domain. 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. Then for testing, the polarity labels of the target domain aspects are added in order to evaluate the performance of DAT-LCR-Rot-hop++.

The reviews of the data sets are divided in single sentences, consisting of an aspect and sentimental context words, which are used to classify the polarity. The sentences are encoded to an XML representation, shown in Figure 3.1. As one can see, the three aspects are “log on”, “WiFi connection”, and “battery life”, while the context words are “fast”, “speedy”, and “long”. Then for each aspect the polarity is defined as either positive, neutral, or negative. The last information in the code line refers to the

positional index of the aspect in the sentence, which will not be used in this research.

```
<sentence id="996:1">
        <text>I am pleased with the fast log on, speedy WiFi connection and the long
        battery life (>6 hrs).</text>
        <aspectTerms>
           <aspectTerm term="log on" polarity="positive"  from="27" to="33"/>
           <aspectTerm term="WiFi connection" polarity="positive"  from="42" to="57"/>
           <aspectTerm term="battery life" polarity="positive"  from="71" to="83"/>
        </aspectTerms>
</sentence>
```

Figure 3.1 The XML representation of the document text

In terms of data pre-processing, the same approach is applied as introduced by Wang et al. (2016) and Zheng and Xia (2018). The implicitly opinionated review sentences contain a sentiment, but the aspect term is missing. This makes it impossible to perform ABSA, using the chosen state-of-the-art 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 lines as the implicitly opinionated sentences are removed from the datasets.

In this paper, our proposed model is concurrently trained on one source and one target domain to obtain domain-invariant features. In total, results will be presented for 6 different domain combinations, being restaurant-laptop, restaurant-book, laptop-restaurant, laptop-book, book-restaurant, and lastly book-laptop. Take for instance the restaurant-laptop model, it means that the restaurant data set is the source domain and the laptop data set equals the target domain. As a consequence, the restaurant observations both have a domain and polarity class, while the laptop aspects only have a domain label for training. When training is finished, the test instances of the laptop domain with sentiment labels are fed into the model. The performance of DAT-LCR-Rot-hop++ is analysed according to its predicting sentiment accuracy of the target test aspects.

3.1 Restaurants

The restaurants SemEval 2014, introduced by Pontiki et al. (2014), contains 4722 different aspects, each classified as either positive, neutral, or negative. As one can see in Table 7.1 in Appendix A, the training set consists of 3600 observations and the testing set include 1122 instances. The amount of positive reactions is significantly higher than both the neutral and negative sentiments for both the training and test set with a percentage of 60.1% and 65.1%, respectively. In total, 2.2% of the aspects are removed due to conflicting sentiments, whereas none are excluded as a result of implicit aspect targets.

3.2 Laptops

The laptops domain is also retrieved from the SemEval 2014 data set (Pontiniki et al., 2014). The set is split into 2250 training observations and 701 test instances. The percentage of positive sentiments is lower compared to the restaurant domain. Overall, this specific data set seems to be the most balanced

in terms of polarities. 2.0% of the data is discarded because of conflicting emotions, and similarly to the restaurants domain, there are no implicit aspects eliminated.

3.3 Books

Last, the books domain, retrieved from the ALT (Álvarez-López et al., 2018), contains 3504 different aspects, split into 2700 training and 804 test observations. It is the only domain that has the highest percentage of neutral emotions, which is 63.1% and 57.1% for the training and test set, respectively. 8.6% of the aspects are excluded in this research on the grounds of implicit target aspects. No conflicting sentiments are removed from this set.

Chapter 4

Methodology

The DAT-LCR-Rot-hop++ model is a combination of two different components. The fundament of the approach is the LCR-Rot-hop++ model, which is described in Section 4.1. This base model is adjusted for cross-domain ABSA using DAT, which is outlined in Section 4.2. This last section explains the structure and implementation of DAT in order to create domain-invariant features.

4.1 LCR-Rot-hop++

The LCR-Rot is introduced by Zheng and Xia (2018) , extended by Wallaart and Frasincar (2019) 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 by Trusca et al. (2020). A visual representation of the neural neutwork is shown in Figure 4.1. Next, the model will be described in detail, step by step.

4.1.1 Contextual word embeddings and bi-LSTMs

First, the sentences are split into three separate parts, consisting of the left context: $[s_1^l, ..., s_L^l]$, target phrase: $[s_1^t, ..., s_T^t]$ =, and right context: $[s_1^r, ..., s_R^r]$. 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 by Devlin et al. (2019). The final word embeddings are calculated by summing the last 4 layers of the BERT model.

$$BERT_i = \sum_{j=9}^{12} H_{i,j}. \tag{4.1}$$

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: $[w_1^l, ..., w_L^l]$, the target word embeddings: $[w_1^t, ..., w_T^t]$, and the right context word embeddings: $[w_1^r, ..., w_R^r]$ are each the input for a three hidden layer bi-LSTM feed-forward neural network, resulting in the hidden states $[h_1^l, ..., h_L^l]$, $[h_1^t, ..., h_T^t]$, and $[h_1^r, ..., h_R^r]$. These

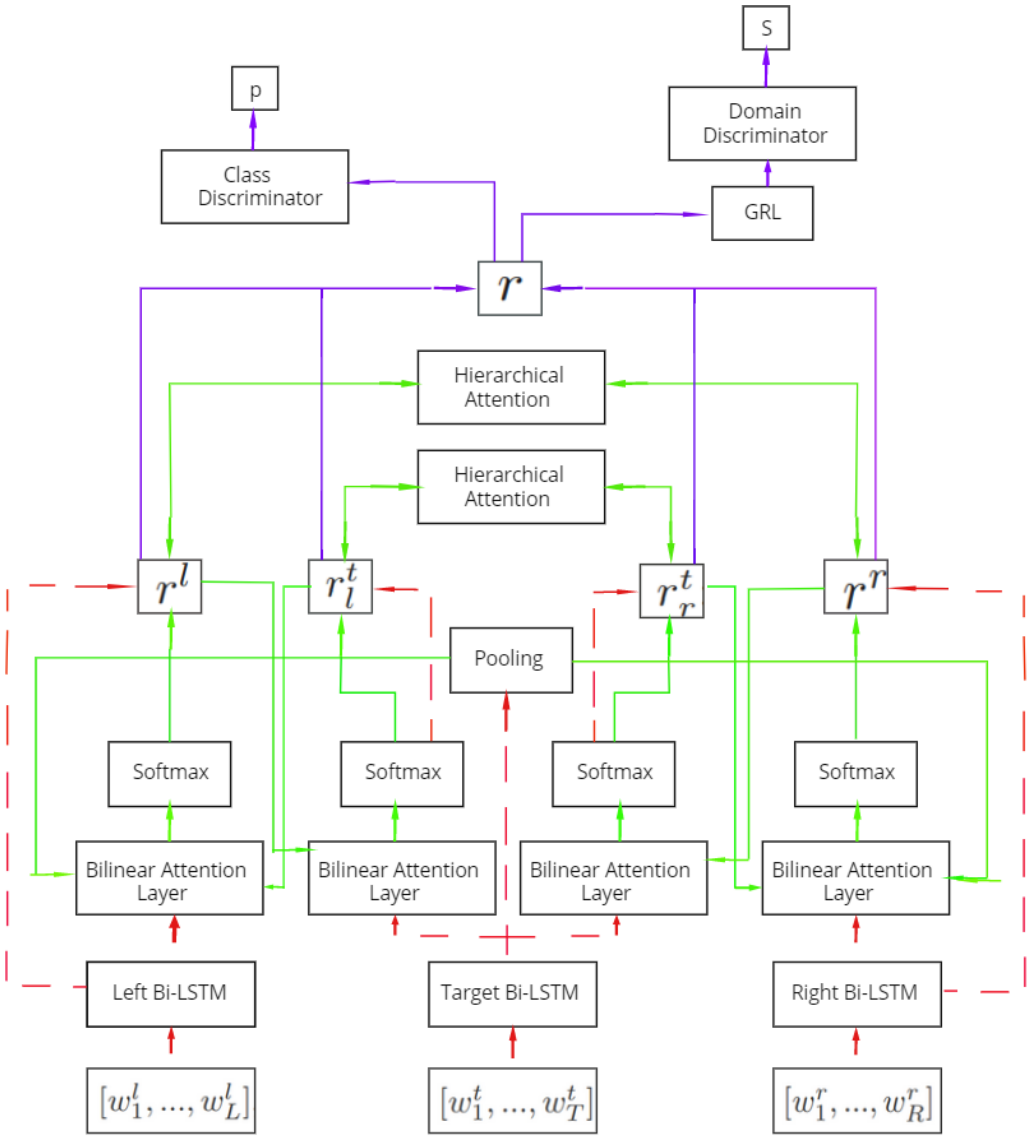


Figure 4.1 A visualisation of the LCR-Rot-hop++ model

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 4.1..

4.1.2 Rotary attention mechanism

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 Equation 4.2. 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 phrases are combined with the h_i^l and h_i^r for the left and right context, respectively.

$$\underset{2d\times 1}{r^t} = pooling([\underset{2d\times 1}{h_1^t}, ..., \underset{2d\times 1}{h_T^t}]) \tag{4.2}$$

A bilinear attention score f, see Equation 4.3, 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.

$$f(\underset{1\times1}{h_i^l}, r^t) = tanh(\underset{1\times2d}{h_i^l} \times \underset{2d\times2d}{W_c^l} \times \underset{2d\times1}{r^t} + \underset{1\times1}{b_c^l}), \tag{4.3}$$

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, all for $i = 1, ..., L$.

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:

$$\alpha_i^l = \frac{exp(f(h_i^l, r^t))}{\sum_{j=1}^{L} exp(f(h_j^l, r^t))}. \tag{4.4}$$

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

$$\underset{2d\times1}{r^l} = \sum_{i=1}^{L} \underset{1\times1}{\alpha_i^l} \times \underset{2d\times1}{h_i^l}. \tag{4.5}$$

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 of r^r and h_i^t. This results in the context2target vectors, r_l^t and r_r^t:

$$\underset{2d\times1}{r_l^t} = \sum_{i=1}^{T} \underset{1\times1}{\alpha_i^{t_l}} \times \underset{2d\times1}{h_i^t}, \tag{4.6}$$

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 left context and the target phrase's hidden output.

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 Equation 4.2 is skipped, because the newly calculated r_l^t and r_r^t are the input to Equation 4.3-4.5. This whole procedure is repeated three times as was decided to be optimal by Wallaart and Frasincar (2019). It is shown by the light green arrows in Figure 4.1.

4.1.3 Rotary hierarchical attention mechanism

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:

$$f(\underset{1\times1}{v^i}) = tanh(\underset{1\times2d}{v^i} \times \underset{2d\times1}{W_h^c} + \underset{1\times1}{b_h^c}), \tag{4.7}$$

where W_h^c is the weight matrix of the hierarchical layer for the context partition and b_h^c represents the bias term for the context 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 Equation 4.8:

$$\alpha^i = \frac{exp(f(v^i)}{exp(f(v^{i_1})) + exp(f(v^{i_2}))}, \tag{4.8}$$

after which the representations are updated:

$$\underset{2d\times 1}{v^i} = \underset{1\times 1}{\alpha^i} \times \underset{2d\times 1}{v^i}, \tag{4.9}$$

This procedure is also rerun multiple times, which is visualised by the green arcs!!!!. Last, the four vectors are concatenated into one single vector, $r = r^l; r_l^t; r_r^t; r^t$ with dimensions $8d \times 1$ and then passed into a Multi-Layer Perceptron (MLP), which uses a softmax function to predict the polarities:

$$p = softmax(W_M \times v + b_M), \tag{4.10}$$

where W_M and b_M are the weight and bias matrix of the MLP, respectively.

4.2 DAT

A GAN, as introduced by Goodfellow et al. (2014), 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 to a vector representation that is ought to capture the important characteristics of the sentence. The other two extra elements are the generator and discriminator. You can see the generator as a painting fraud, aiming to make the painting look as real as possible, while the discriminator is the painting expert, trying to determine whether or not the painting is real. So, the generator is striving to fool the discriminator. As you can imagine, both operate in an adversarial manner.

Ganin and Lempitsky (2015) have layed the fundament of a method that is now known as DAT. Their model is a specification of the GAN network, previously described. They introduced an approach in deep learning that is able to adapt on target domains without any labeled target data. This is done through generating deep features that are discriminative for the main learning classifying task by using the labeled sentiments of the source domain. At the same this method ensures that these representations are invariant to shifts between the source and target domain in order to be domain in-discriminative by applying the domain class of both the source and target domain. The proposed DANN solution is revisited by Ganin et al. (2016), who provide a more detailed and elaborate description of the mathematics behind the system.

The change to the GAN network in order to be conform to a DANN model reduces to the fact that for creating cross-domain features, the generator is excluded and replaced by a GRL. This GRL also thrives to make the task of the domain discriminator as hard as possible, which is the direct connection to the GAN. In the DAT-LCR-Rot-hop++, the loss of the domain discriminator is passed through the GRL, which reverses the gradient before back-propagation into the feature extractor. This causes the hidden layers of the LCR-Rot-hop++ to react by constructing features which will not be recognized 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. To refer this back to the painting example, it means that the expert cannot say correctly whether the painting is fake or real.

4.2.1 Structure

The main difference to the original LCR-Rot-hop++ model is the removal of the MLP output layer and its replacement by a domain adversarial component. After the context and target representations are concatenated into r, produced by the feature extractor, this vector is passed into two standard feed-forward MLPs, which are the class discriminator and the domain discriminator. First, the domain discriminator aims to correctly classify the domain of r with Equation 4.11. The predicted domain is given by s. The possible domains are either the source, S, or target, T, domain, making it a binary problem with $d = 0$ for source and $d = 1$ for target actual domain labels. Next, the class discriminator uses a softmax function in Equation 4.12 to compute the probabilities of the sentiment of the aspect, resulting in a 1×3 output vector, p. The polarity that has the largest probability will be chosen as the final sentiment. In both equations, y represents the actual polarity label, x the input sentence, and d is the real domain $\in \{0, 1\}$. The sigmoid function is used for the domain prediction, because it shows good performance for examining binary cases and is applied by multiple researches in domain discriminators (Hong et al., 2018) (Zhang et al., 2018). The DAT component is visualised by the dark purple solid arrows in Figure 4.1.

$$s = sigmoid(W_d \times x + b_d), \tag{4.11}$$

where W_c and b_c represent the weight and bias term of the domain discriminator, respectively.

$$p = softmax(W_c \times x + b_c), \tag{4.12}$$

where W_c and b_c are the weight and bias term of the class discriminator, respectively.

In general, the task of an ABSA model is to minimise the predicting loss. In this paper, this means reducing the error term of both the domain discriminator, noted as $L_d(\theta_f, \theta_d)$, and the class discriminator, defined as $L_c(\theta_f, \theta_c)$. Here, θ represents the parameters of the feature extractor (LCR-Rot-hop++), the domain discriminator, and the class discriminator, defined by the underscores f, d, and c, respectively. Hence, the objection function to optimise is:

$$\min_{\theta} L_{c,d}(\theta_f, \theta_c, \theta_d) = L_d(\theta_f, \theta_d) + L_c(\theta_f, \theta_c). \tag{4.13}$$

However, as described in Section 4.2, the GRL tries to fool the discriminator. After the domain is predicted as either S or T 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 in-discriminative,

which is the primary objective of the GRL. This process leads to the following adjusted loss function:

$$\min_{\theta} \; L_{c,d}(\theta_f, \theta_c, \theta_d) = -\lambda L_{c,d}(\theta_f, \theta_c, \theta_d) + L_c(\theta_f, \theta_c), \tag{4.14}$$

where L_d is:

$$L_d(\theta_f, \theta_d) = \sum_{i=1}^{N} d_i * log(s_i) + \pi_d * ||\theta_d||^2, \tag{4.15}$$

and L_c is:

$$L_c(\theta_f, \theta_c) = \sum_{i=1}^{n} y_i * log(p_i) + \pi_c * (||\theta_f||^2 + ||\theta_c||^2). \tag{4.16}$$

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. Lastly, 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 and only the source instances are passed into the class discriminator.

As one can see, this function is now minimised when the loss of the domain discriminator is maximised. This min-max situation resolves to:

$$\hat{\theta_d} = \arg\max_{\theta_d} L_{c,d}(\hat{\theta_f}, \hat{\theta_c}, \theta_d) \tag{4.17}$$

$$(\hat{\theta_f}, \hat{\theta_c}) = \arg\min_{\theta_f, \theta_c} L_{c,d}(\theta_f, \theta_c, \hat{\theta_d}) \tag{4.18}$$

At this saddle point, the parameters of the domain discriminator, θ_d, minimise the loss of its discriminator, $-\lambda L_d(\theta_f, \theta_d)$, to guarantee accurate domain prediction. So instead of general descending the gradient, in this case ascending gradient is applied. Secondly, θ_c and θ_f are computed to optimise Equation 4.14. The weights and biases of these components are estimated in such a way that they minimise the sentiment prediction loss, hence provide label discriminative features, and maximise the domain classification error, thereby producing domain-invariant features. The hyperparameter λ regulates the balance and trade-off between both goals.

The original DANN paper (Ganin & Lempitsky, 2015) implements Stochastic Gradient Descent (SGD) optimisation. However, the state-of-the-art image classifying model proposed by Mauro et al. (2021) shows that utilising the faster momentum method (Liu & Belkin, 2017) instead of SGD also produce accurate results. Whereas the SGD zig-zaggs its way down to the optimal point, the momentum gradient descender applies a technique which can be seen as pushing a ball down the hill. As a result, it reduces oscillation and gains faster convergence. For this reason, the momentum optimiser is used in this work. In each iteration, the parameters of the neural network will be updated to this method accordingly:

$$v_t \longleftarrow \gamma * v_{t-1} + \eta * \nabla_{\theta_k} L(\theta_k) \tag{4.19}$$

$$\theta_k \longleftarrow \theta_k - v_t. \tag{4.20}$$

Here, the hyperparameters are the learning rate, η, and 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 portrays the corresponding loss function.

4.2.2 Implementation details and training procedure

Normally, when applying DANN, the sample sizes of S and T are similar. However, due to the fact that the three input data sets have different number of aspects, this requirement is removed. This causes the model to be able to implement different sizes of source and target data, which increases the overall applicability and robustness of the network. As stated, 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 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. During testing, the target domain test instances are passed through the class discriminator in order to predict their sentiment and compare it with the actual polarity to obtain a label classifying accuracy. The performance of the DAT-LCR-Rot-hop++ is evaluated based on this testing accuracy.

The weights and biases of the domain discriminator, the feature extractor, and the class discriminator are improved using the combined loss function, given by Equation 4.14. This Equation includes the $-\lambda$ multiplication in order to create sentiment discriminative and domain indiscriminate features. However, as previously mentioned, the domain discriminator uses ascending gradient to maximise this loss function, whereas the feature extractor and the class discriminator minimise it. The exact training procedure is shown in Algorithm 1.

The other hyperparameters besides the λ in the 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 $k = f$ for the feature extractor and class discriminator. First, η_k determines the rate at which the momentum optimiser converges. A high value could prevent other mechanisms to reach their optimum because one of the components in the system is fully optimised and does not let other components further converge, while a low value could withhold the model from attaining its optimum and increases computation time significantly. In addition, γ_k determines the influence of past gradient values on the current instance. Furthermore, π_k reduces overfitting. 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. A value of 1.0 means no dropout, while a value of 0.0 clears all output.

Wallaart and Frasincar (2019) and Trusca et al. (2020) both obtain optimal hyperparameter values for their models. Because the dropout rate does not differ between both papers, 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). This approach replaces the distribution of the initial observations with a non-parametric distribution by applying a threshold for which it decides whether or not the observation belongs to a certain density (Bergstra et al., 2020). π_d and π_f are kept at the same value in this paper, but they are allowed to differ for future research.

Algorithm 1: Training procedure of Domain-Adversarial Learning

acc_t = training sentiment accuracy at epoch t

$\epsilon = 0.50\%$

while $max(acc_{t-1}, acc_{t-2})$–$acc_{t-3} > \epsilon$ **do**

  for *each epoch* **do**

    for *each iteration i* **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. denotes the source domain batch size and is the total batch size.
- Feed input into feature extractor to obtain word representations
- 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 features, $\hat{\theta_d}$, with descending gradient:

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

    end

  end

end

As in the research performed by Wallaart and Frasincar (2019) and Trusca et al. (2020), the dimension of the word embeddings, $1 \times d$, is equal to 1×768. For convenience, the number of nodes in the bi-LSTMs, hierarchical and bilinear attention layer are the same as in (Trusca et al. ,2020). These are 300, 300, and 600, respectively. The number of hidden layers and cells in both the class and domain discriminator are optimised by TPE. For simplicity, both discriminators consist of the same amount of hidden layers and neuron cells. The proposed number of layers is either 1 or 2. This results from the fact that Heaton (2017) states that generally one layer should be capable of approximating any function with a finite feature mapping. At the same time, Lipmann (1987) says that it is sometimes more efficient to use two layers, which he theoretically proves to be sufficient for classifying deep learning problems of any shape. Furthermore, Uzair and Jamil (2020) argue that having too many layers in a neural network could lead to overfitting on the training data, whereas too little neurons and layers can result in underfitting. If applying 1 layer produces the highest sentiment accuracy, the number of cells will be set to 2400 as in the MLP of Trusca et al. (2020). If 2 layers results to be optimal, the number of neurons will be 2400 and 1200 or 2400 and 600 for the two layers, consecutively. Each weight is initialised randomly using a normal distribution with mean 0. On top of that, the biases are set to zero at the start.

After the hyperparameters are initialised, the DAT-LCR-Rot-hop++ is trained on the training set. The sentiment accuracy of the validation set is used to decide which combination of parameter coefficients achieves the best performance. The number of hidden layers are also incorporated in the parameter optimisation. In the best case, these values will be fine-tuned by running all the possible combinations of these parameters for an infinite amount of epochs or until the stopping condition of Algorithm 1 is reached. However due to time constraints, this is not possible. As a consequence, we have 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 includes 50 epochs.

The hyperparameter fine-tuning occurs twice. In the first step, λ is excluded. The reason for this is 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 a value of 1.0 (Ganin et al., 2016) in order to find the optimal values for the other parameters. After the influence of λ is analysed, 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.

Chapter 5

Evaluation

In the following chapter, we first describe the influence of λ on the performance of DAT-LCR-Rot-hop++. Then, the results for the final optimisation are shown.

5.1 Impact of λ

First, the optimal number of hidden layers and neurons together with the coefficients for the hyperparameters are computed. The results are shown in 5.1. These values are used to determine the effect of λ. As stated, the number of tested runs is 15 and the number of epochs 50 due to time constraints, so it could be possible that another mix of coefficients achieve better performance.

Table 5.1 Possible values for hyperparameter and structure optimisation

Hyperparameter	**Possible values**	**res-lap**	**res-book**	**lap-res**	**lap-book**	**book-res**	**book-lap**
lr_d	[0.005, 0.01, 0.03, 0.07]	0.03	0.03	0.005	0.01	0.03	0.03
$lr_{c,f}$	[0.005, 0.01, 0.03, 0.07]	0.005	0.03	0.01	0.005	0.03	0.03
mom_d	[0.80, 0.85, 0.90]	0.80	0.90	0.85	0.90	0.90	0.80
$mom_{c,f}$	[0.80, 0.85, 0.90]	0.90	0.80	0.90	0.85	0.80	0.85
$l2-term$	[0.0100, 0.0010, 0.0001]	0.001	0.0001	0.001	0.0001	0.001	0.0001
structure	[2400-0, 2400-600, 2400-1200]	2400-1200	2400-0	2400-600	2400-0	2400-600	2400-600

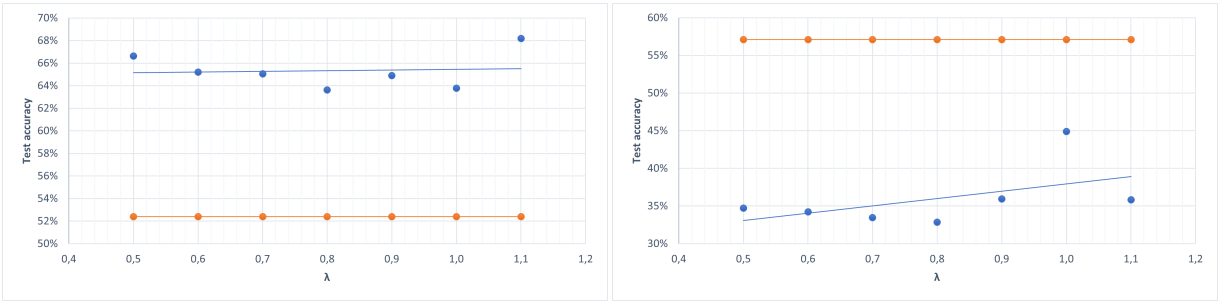
The DAT-LCR-Rot-hop++ is run for 7 incrementing values of λ, starting from 0.5 up until 1.1 for each domain combination. The impact of the balance hyperparameter λ is visualised by the six graphs that follow this section. In these graphs, the dark blue line represents the labeling accuracies of the test set of the target domain, while the light orange line shows the base performance when the majority group of the test sample was selected. The model used a maximum of 50 epochs. As a consequence, some models jumped up and down and were not close to converging yet, so some results might be inconclusive. The outcomes are shown in more detail in Table 8.1 in Appendix B. The first sections introduce some general interesting outcomes. Next, we go into more detail about each figure individually.

When analysing the graphs, 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 consumer 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 domain are taken from the SemEval 2014 (Pontiki et al., 2014) dataset while the books domain is retrieved from the ALT 2019 (Álvarez-López et al., 2018). First of all, these datasets share 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. In addition, the language might have developed throughout these 5 years, which causes people to use different words in sentences. Third and last, the fraction of neutral aspects in the book data test set is significantly higher than the training set of both the restaurant and laptop domain with a percentage of 63.1, 17.7, and 19.8. This causes extreme emotional phrases, for example "awesome", to appear 5 times in the book domain and 30 and 16 times in the laptop and restaurant domain. As a result, there is more overlap between the restaurant and laptop domain. 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 book as a target domain is worse than applying book as a source domain. The low accuracy of restaurant-book and laptop-book is due to the high percentage of neutral aspects in the book data test. Having a disproportionate training set causes the neurons to react by predicting the sentiment that occurs most often, especially in the earlier iterations. This logically produces the highest possible beginning classifying accuracy. So, for both the restaurant and laptop domain this results in starting with improving the positive predicting abilities and then continuing with the negative polarities. The neurons will be trained the latest to be able to predict neutral emotions. As a consequence, the extreme neutral percentage of the book target domain requires more complex training and will lead to lower performance. On the other hand, using book as source domain does lead to an acceptable performance. We come up with one reason for this outcome. The book-restaurant and book-laptop model both start off with predicting the neutral aspects due to the high fraction of neutral aspects in the ALT 2019 (Álvarez-López et al., 2018) set. Not surprisingly, most neutral aspects are correctly classified in the book-restaurant and book-laptop combinations as shown in 8.1. The average neutral accuracy for the book-restaurant model is 65% and 81% for the book-laptop combination. Next, the DAT-LCR-Rot-hop++ focuses on the second largest polarity percentage, the positive sentiments. Because both the restaurant and laptop domain mostly consist of positive aspects, this results in a good performance. The drawback of this is the bad score for the negative polarities, which have an accuracy of 3.2% and 0% for the book-restaurant and book-laptop domain, respectively.

When looking at Figure 5.1a, we observe a scattered graph with an almost flat regression line. Accordingly, the coefficient of the OLS slope is 0.61, which means that increasing λ with 1.0 increases the testing accuracy with 0.61%. The same holds for Figure 5.2a, which has a slope of 2.04. Apparently, λ does not have a strong effect on both restaurant-laptop and laptop-restaurant. One reason for this could be the described overlap between both domains. This results in a low difference between both sets, thereby decreasing the difficulty of the cross-domain task and hence, making λ less important. 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 for which the observation at $\lambda = 0.9$ appears to be an outlier.



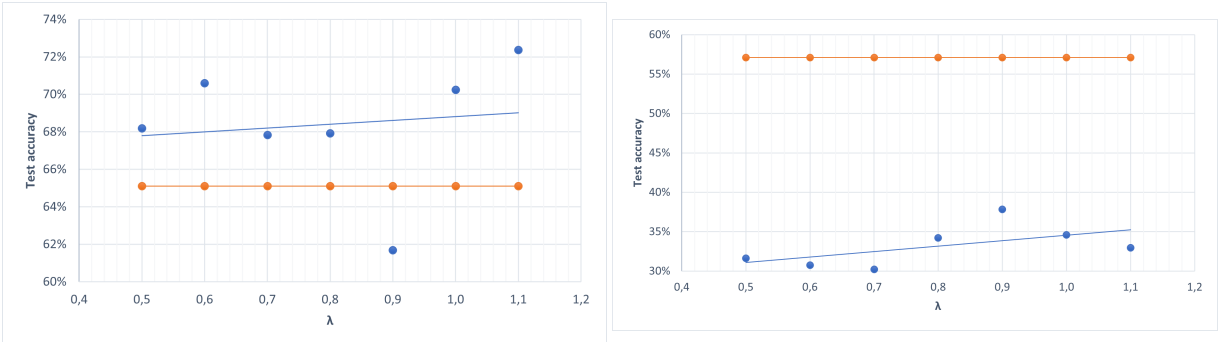
(a) restaurant-laptop

(b) restaurant-book

Figure 5.1 Classifying test accuracy with restaurant as source domain for different values of λ

In contrast to the flat regression lines of the restaurant-laptop and laptop-restaurant, restaurant-book shows a clear positive linear trend with a slope of 9.73. This might be attributed to the outlier at λ is 1.0. However, after removing the outlier, the slope still exceeds 4%. This is a direct proof of the influence of λ. The same applies to the graph in Figure 5.1b, which has a slope of 6.91.

As stated, employing book as the target domain produces a poor labeling accuracy. The effectiveness of the model depends on its ability to create correct word representations for the neutral aspects, because the percentage of neutral polarities is the highest. Especially, the positive outlier in Figure 5.1b is an observation that shows the effect of the disproportionate data sets. Only during this run, the DAT-LCR-Rot-hop++ was capable of predicting neutral sentiments correctly, which results in a neutral accuracy of 26%. For the six other values of λ, this outcome has a maximum of 5%. The competence to classify the neutral aspects precisely immediately leads to a significantly better performance with an accuracy of 45% as compared to a maximum of 36%. However, this only happened once. Also for the laptop-book model in Figure 5.2b, the maximum neutral accuracy after all the epochs is 4%. As a consequence, the test accuracy moves around 35%, which is far below the benchmark of 57.1%. The disproportionate distributions of the SemEval 2014 (Pontiki et al., 2014) and ALT 2019 (Álvarez-López et al., 2018) set make it hard for the model to create features which are good for predicting all three sentiments. This causes the DAT-LCR-Rot-hop++ to focus on the major two polarities, which are the positive and negative sentiments when training on the restaurant and laptop domain, resulting in low scores for the restaurant-book and laptop-book combination.

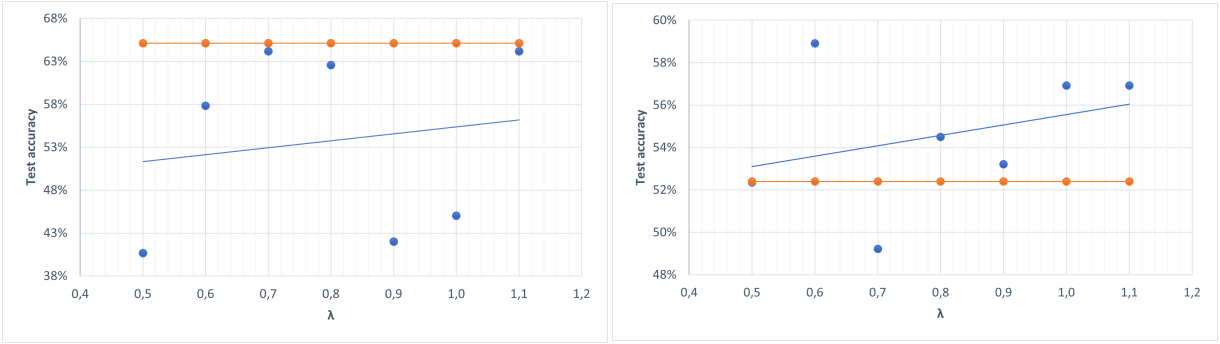


(a) laptop-restaurant

(b) laptop-book

Figure 5.2 Classifying test accuracy with laptop as source domain for different values of λ

Similar, to restaurant-book and laptop-book, both book-restaurant and book-laptop in Figure 5.3a and 5.3b, respectively, provide an ascending line. The book-restaurant has a slope value of 7.04 and the book-laptop has a slope of 4.88. However, the data points in Figure 5.3a are scattered, causing a standard deviation of 10%. Therefore, this positive relationship might be questioned. Still, this outcome supports the statement from the introduction, which says that a higher value for λ should improve the cross-domain applicability of the features. If there is enough difference between the source and target domain in terms of overlap and distribution, increasing the balance hyperparameter λ forces the model to make cross-domain features by magnifying the importance of the loss of the domain discriminator.



(a) book-restaurant

(b) book-laptop

Figure 5.3 Classifying test accuracy with book as source domain for different values of λ

On the other hand, for each run, the accuracy of the domain discriminator converges relatively close to the desired 0.50, which equals a random walk model taking into account the uneven sample sizes, after approximately 30 epochs. This result implies that the DAT-LCR-Rot-hop++ is able to create features, which are hard to label correctly by the domain discriminator, for all values of λ. So, the argument that a higher λ directly increases the domain-invariance of the features of the neural network might be too strong , but λ does affect the performance of the model. Therefore, it should be treated as one of the hyperparameters of the model.

We also believed that a higher λ would cause a lower training accuracy. Because you make the loss of the domain discriminator more prominent, the influence of the class discriminator is decreased. This statement is proven wrong by the results in Table 8.1. The accuracy of the training set never differed further than 3% from its mean while changing λ. In addition, the movements did not follow one specific direction. Apparently, λ does not have an influence on the performance of the labeling accuracy of the source domain on which the class discriminator is trained.

5.2 Final optimisation

Next, the possible values for the hyperparameter fine-tuning for the final optimisation with 200 iterations are defined. These values are chosen after considering the results from testing the impact of λ. The optional and optimal values are shown in Table 5.2. Each domain combination will be tested for the final prediction using these parameter settings.

Table 5.2 Possible values for hyperparameter optimisation

Hyperparameter	**Possible values**	**res-lap**	**res-book**	**lap-res**	**lap-book**	**book-rest**	**book-lap**
lr_d	[0.005, 0.01, 0.03]	0.01	0.03	0.03	0.01	0.01	0.03
$lr_{c,f}$	[0.005, 0.01, 0.03]	0.005	0.005	0.03	0.01	0.005	0.03
mom_d	[0.80, 0.85, 0.90]	0.90	0.85	0.80	0.90	0.85	0.90
$mom_{c,f}$	[0.80,0.85, 0.90]	0.90	0.90	0.85	0.85	0.85	0.85
$l2-term$	[0.01, 0.001, 0.0001]	0.001	0.001	0.001	0.001	0.001	0.0001
structure	[2400-0, 2400-600, 2400-1200]	2400-600	2400-0	2400-600	2400-600	2400-0	2400-600
λ	[0.6, 0.8, 1.1]	1.1	0.8	1.1	0.6	1.1	0.6

The final results are shown in Table 5.3. The outcomes do not differ much from the observations in Table 8.1. As expected, the accuracies improve for each source and target domain model as compared to the previous run with 50 epochs. The training label accuracy increases from 84% up until 92% for the book-laptop domain. At the same time, the maximum testing accuracy of 80% for the restaurant-laptop is a 9% improvement from the 71% in Table 8.1. In addition, the ratios of correctly predicted polarities follow the same distribution.

Table 5.3 Test accuracies for final training procedure of CLRH++ model

	test acc	**train acc**	**max acc**	**base acc**	**positive acc**	**neutral acc**	**negative acc**
rest-lapt	77%	87%	80%	52%	94%	38%	83%
rest-book	37%	88%	48%	57%	71%	9%	88%
lapt-rest	74%	83%	78%	65%	90%	9%	85%
lapt-book	42%	90%	52%	57%	86%	11%	72%
book-rest	72%	88%	76%	65%	78%	81%	40%
book-lapt	60%	92%	66%	52%	75%	79%	0%

Base acc equals the labeling accuracy if the majority class of the test set of the target domain is predicted.

The first major difference that attracts our attention is the relatively large percentage of correctly specified neutral aspects for the restaurant-laptop combination. It is 3.8 times as high as the average for the runs when evaluating the impact of λ. Furthermore, its performance with respect to the positive sentiment polarities and negative aspects is excellent, which results in a total test accuracy of 77%. A 77% labeling accuracy might not come across as a good predicting score if you compare it with other state-of-the art models. However, first of all, it beats the result of van Berkum et al. (2021) by 5%. More importantly, the DANN approach does not require any labeled target data. The relevance of the second advantage should not be underestimated when comparing it with other researches because the ability to correctly label sentiments of other unknown domains is valuable and reduces labeling computation costs significantly. Specifically, the outcomes for the book-restaurant testing are promising. Both domains are not closely related in terms of sentiment distribution, but the model achieves an encouraging test accuracy of 72%, which is an improvement of 8 per cent as compared to the maximum value after the 50 iterations. In addition, the fraction of correctly labeled sentiments is well-divided. Instead of one polarity that drags the results, each contributes.

On the other hand, one cannot ignore the low scores for the restaurant-book and laptop-book combination with a testing accuracy of 37% and 42%, respectively. Multiple causes for this unfortunate outcome are given in Section 5.1. Especially, the disproportionate fraction of neutral aspects seems to be the main issue here. The excellent classifying scores for the positive and negative sentiments are offset by the terrible performance for the neutral aspects. Furthermore, the difference between the maximum testing accuracy during the 200 epochs and the final percentage of correct predictions is also significantly higher as compared to the other models. Whereas dividing the maximum accuracy by the final testing accuracy has an average factor of 1.06 for the four other domain combinations, this statistic is 1.30 for the restaurant-book and 1.24 for the laptop-book model. In addition, overfitting occurs for the restaurant-book and laptop-book model. We analysed multiple epochs one by one. 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 into both directions, but never really improving. After some time, the accuracy start to drop. This can indicate overfitting. The model becomes too much specified towards the information of the training set. This also gives reason to believe that the DANN is not able to fully create domain-invariant features. As mentioned, the domain discriminator's accuracy decreases significantly until a value close to 0.5. This would mean domain in-discriminative weights and biases because the domain discriminator cannot detect to which domain the word representations belong. However, at some point, only the class accuracy of the training set continues improving. This implies that the features are not fully cross-domain applicable. The DAT-LCR-Rot-hopp++ is not capable of reverting the overfitting mechanism. For this reason, we believe it is important for future research to investigate the essential role that data balancing plays in DANN and whether there is a maximum to the transfer learning capabilities of a DANN.

Chapter 6

Conclusion

The prominent role of digital media increases the relevance of ABSA. It provides important applications and solutions to current problems. Still, a single task can require the model to perform well on multiple domains. In the optimal case, you train the model on the new domain with labeled data in order to achieve the best results. However, for several domains these annotated observations do not exist. Since obtaining labeled target data is extremely costly in terms of money and labour, new models should be developed that can be employed on a variety of domains, a concept known as transfer learning. Several researches tackle the problem in their own way, ranging from masking (Yuan et al., 2021) to freezing (Chen, 2020). Ganin et al. (2016) introduce a method known as DANN, which is a specification of the GAN as defined by Goodfellow et al. (2014). The major benefit of this approach is the fact that it does not need any labeled target data at all.

Our work builds on the approach of Trusca et al. (2020). Their state-of-the-art LCR-Rot-hop++ structure forms the basis of our proposed DAT-LCR-Rot-hop++, which adds an adversarial component. This layer should improve the cross-domain predicting performance of the model. It consists of a domain discriminator, a class discriminator and a GRL. The GRL reverses the loss of the domain discriminator before back-propagation, which enforces the earlier layers to generate domain-invariant features. At the same time, the class discriminator is trained on the labels of the source domain, which results in label discriminative features.

The effect of the balance hyperparameter λ is present for the models that include the book domain. Increasing λ improves the performance of the DAT-LCR-Rot-hop++ if there is enough difference between the source and target domain. But, at the same time, the domain discriminator is not able to correctly predict the domain of the input for each value of λ. So the features are also domain-invariant for low values of λ. Therefore, λ is treated as a hyperparameter for the final optimisation.

The final results show that the DANN is not able to correctly predict the sentiments of the restaurant-book and laptop-book models. The benchmark of predicting the majority class of the test set is not reached for both domain combinations. This is the outcome of multiple factors of which the most essential one is the disproportionate data set in terms of polarity distribution of the three domains. The high percentage of neutral aspects cause the model to perform poorly. Unfortunately, it lacks strength

to transform the weights and biases in more domain-invariant features.

On the other hand, the accuracy score for the restaurant-laptop, laptop-restaurant, and book-restaurant all exceeded 72%. So in half of the cases, the DAT-LCR-Rot-hop++ is able to properly classify polarities. But, it depends on which combination of domains is used. A part of the 77% and 74% test accuracy can be attributed to the similar distribution of the polarities in the restaurant and laptop data sets. The high positive polarity percentage of both the restaurant and laptop data set cause the model to start off with predicting a positive label for all aspects, which results in a 65% test accuracy for the restaurant domain and a 52% for the laptop domain. Afterwards, the neural network improves it positive classifying abilities and moves its attention to the second largest fraction, which is the negative sentiment. The same story applies to the book-restaurant model. This model begins with predicting a neutral label and then continues with the positive polarity. However, for both the restaurant-laptop and book-restaurant, the DAT-LCR-Rot-hop++ is able to label all three sentiments. Therefore, the stated results cannot be completely attributed to the similar data set distributions and overlap, but also to the proper performance of our proposed model.

In order to further improve our method, I propose three adaptions. The first one covers the difficulty of predicting three kinds of classes. We would like to investigate the performance of the model for only the binary case. Classifying neutral aspects appears to be a harsh task for the neural network, looking at the results. Furthermore, it is important to analyse the effectiveness of the LCR-Rot-hop++ if the data sets have a similar polarity distribution. If these results prove to be better, it can be interesting to examine the crucial role of data balancing in DANN. Our last idea includes transforming the complete LCR-Rot-hop++ component. Due to its complexity and large amount of layers with corresponding weights and biases, the model becomes overfitted on the training data. Because all the parameters of the feature extractor are more influenced by the combined domain and label loss of the source data than only the domain loss of the target data, these coefficients are adjusted to the needs of the source domain. The final goal is to predict polarities for the target domain, so we would rather observe a less complicated feature extractor and maybe a more elaborate domain and class discriminator in order to make the function of the GRL more influential.

Bibliography

Álvarez-López, T., M. Fernández-Gavilanes, E. Costa-Montenegro, and P. Bellot (2018). "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)*. Vol. 10859. LNCS. Springer, pp. 3–14.

Bergstra, J., R. Bardenet, Y. Bengio, and B. Kégl (2011). "Algorithms for Hyper-Parameter Optimization". In: *25th Annual Conference on Neural Information Processing Systems (NIPS 2011)*. Curran Associates, pp. 2546–2554.

Berkum, S., S. Van Megen, M. Savelkoul, and P. Weterman (2021). *Fine-Tuning for Cross-Domain Aspect-Based Sentiment Classification*. Tech. rep. Theoretical Report of Erasmus University Rotterdam.

Bird, J., D. Faria, J. Kobylarz, and A. Ekárt (2020). "Cross-domain MLP and CNN Transfer Learning for Biological Signal Processing: EEG and EMG". In: *IEEE Access* 8, pp. 54789–54801.

Chapman, B. (2021). *GameStop: Reddit Users Claim Victory as $13bn Hedge Fund Closes Position, Accepting Huge Losses*. URL: `https://www.independent.co.uk/news/business/gamestop-share-price-reddit-hedge-fund-melvin-capital-b1793543.html`.

Chen, Y., Z. Tong, Y. Zheng, H. Samuelson, and L. Norford (2020). "Transfer Learning with Deep Neural Networks for Model Predictive Control of HVAC and Natural Ventilation in Smart Buildings". In: *Journal of Cleaner Production* 254 (119866).

Ciresan, D., U. Meier, and J. Schmidhuber (2012). "Multi-column Deep Neural Networks for Image Classification". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*. Vol. 1. IEEE Computer Society, pp. 3642–3649.

Dechter, R. (1986). "Learning while Searching in Constraint-Satisfaction-Problems". In: *5th National Conference on Artificial Intelligence (AAAI 1986)*. Morgan Kaufmann, pp. 178–185.

Devlin, J., K. Chang, K. Lee, D. Huang, and K. Toutanova (2019). "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)*. ACL, pp. 4171–4186.

Fan, C., Y. Sun, F. Xiao, J. Ma, D. Lee, J. Wang, and Y. Tseng (2020). "Statistical Investigations of Transfer Learning-based Methodology for Short-Term Building Energy Predictions". In: *Applied Energy* 262, pp. 114–149.

Ganin, Y. and V. Lempitsky (2015). "Unsupervised Domain Adaption by Backpropagation". In: *32nd International Conference on Machine Learning (ICML 2015)*. Vol. 37. PMLR, pp. 1180–1189.

Ganin, Y., E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky (2016). "Domain-Adversarial Training of Neural Networks". In: *Journal of Machine Learning Research* 17, 59:10–59:35.

Glorot, X. and Y. Bengio (2010). "Understanding the Difficulty of Training Deep Feedforward Neural Networks". In: *13th International Conference on Artificial Intelligence and Statistics (ICAIS 2010)*. Vol. 9. PMLR, pp. 249–256.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). "Generative Adversarial Networks". In: *28th Annual Conference on Neural Information Processing Systems (NIPS 2014)*, pp. 2672–2680.

Heaton, J. (2017). *The Number of Hidden Layers*. URL: `https://www.heatonresearch.com/2017/06/01/hidden-layers.html`.

Hendricks, D. (2021). *Complete History of Social Media: Then And Now*. URL: `https://smallbiztrends.com/2013/05/the-complete-history-of-social-media-infographic.html`.

Hong, W., Z. Wang, M. Yang, and J. Yuan (2018). "Conditional Generative Adversarial Network for Structured Domain Adaption". In: *8th IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2018)*. Vol. 12. IEEE, pp. 1335–1344.

Hong, Y., W. Zhou, J. Zhang, Q. Zhu, and G. Zhou (2018). "Self-Regulation: Employing a Generative Adversarial Network to Improve Event Detection". In: *56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*. ACL, pp. 515–526.

Hu, M. and B. Liu (2004). "Mining and Summarizing Customer Reviews". In: *10th International Conference on Knowledge Discovery and Data Mining (KDD 2004)*. ACM, pp. 168–177.

Huang, Z., W. Xu, and K. Yu (2015). "Bidirectional LSTM-CRF Models for Sequence Tagging". In: arXiv preprint arXiv: `1508.01991`.

IJntema, W., F. Goossen, F. Frasincar, and F. Hogenboom (2010). "Ontology-Based News Recommendation". In: *EBDT'10: 2010 International Conference on Extending Database Technology (EBDT/ICDT 2010)*. Vol. 16. ACM International Conference Proceeding Series. ACM, pp. 1–6.

Johansson, F., U. Shalit, and D. Sontag (2016). "Learning Representations for Counterfactual Inference". In: *33rd International Conference on Machine Learning (ICML 2016)*. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR, pp. 3020–3029.

Kamath, S., S. Gupta, and V. Carvalho (2019). "Reversing Gradients in Adversarial Domain Adaption for Question Deduplication and Textual Entailment Tasks". In: *57th Annual Meeting of the Association for Cumputational Linguistics (ACL 2019)*. ACL, pp. 5545–5550.

Karamizadeh, S., S. Abdullah, M. Halimi, J. Shayan, and M. Rajabi (2014). "Advantage and Drawbacks of Support Vector Machine Functionality". In: *2014 International Conference on Computer, Communications, and Control Technology (I4CT 2014)*. IEEE.

Lai, S., L. Xu, K. Liu, and J. Zhao (2015). "Recurrent Convolutional Neural Networks for Text Classification". In: *29th Conference on Artificial Intelligence (AAAI 2015)*. Vol. 29. 1. AAAI Press, pp. 2267–2273.

Larochelle, H., Y. Bengio, J. Louradour, and P. Lamblin (2009). "Exploring Strategies for Training Deep Neural Networks". In: *Journal of Machine Learning Research* 10, pp. 1–40.

LeCun, Y., B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel (1989). "Back-Propagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1 (4), pp. 541–551.

Li, X., L. Bing, W. Zhang, and W. Lam (2019). "Exploiting BERT for End-to-End Aspect-based Sentiment Analysis". In: *5th Workshop on Noisy User-generated Text (W-NUT 2019)*. ACL, pp. 34–41.

Lipmann, R. (1987). "An Introduction to Computing with Neural Nets". In: *IEEE ASPP Magazine* 4 (2), pp. 4–22.

Liu, B. (2020). *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.

Liu, C. and M. Belkin (2020). "Accelerating SGD with Momentum for Over-Parameterized Learning". In: *8th International Conference on Learning Representations (ICLR, 2020)*. OpenReview.net.

Maas, A., R. Daly, P. Pham, D. Huang, A. Ng, and C. Potts (2011). "Learning Word Vectors for Sentiment Analysis". In: *49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*. ACL, pp. 142–150.

Maat, E. De, K. Krabben, and R. Winkels (2010). "Machine Learning versus Knowledge Based Classification of Legal Texts". In: *23rd Annual Conference on Legal Knowledge and Information Systems (JURIX 2010)*. Vol. 223. IOS Press, pp. 87–96.

Mauro, M., V. Mazzia, A. Khalil, and M. Chiaberge (2021). "Domain-Adversarial Training of Self-Attention Based Networks for Land Cover Classification using Multi-Temporal Sentinel-2 Satellite Imagery". In: arXiv preprint arXiv: `2104.00564`.

Meijer, L., F. Frasincar, and M. Trusca (2021). "Explaining a Neural Attention Model for Aspect-Based Sentiment Classification using Diagnostic Classification". In: *36th ACM Symposium on Applied Computing (SAC 2021)*. ACM, pp. 821–827.

Mikolov, T., M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur (2010). "Recurrent Neural Network Based Language Model". In: *11th Annual Conference of the International Speech Communication Association (ISCA 2010)*. ISCA, pp. 1045–1048.

Pan, S. and Q. Yang (2009). "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22 (10), pp. 1345–1359.

Pang, B. and L. Lee (2004). "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts". In: *42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*. ACL, pp. 271–278.

Pang, B., L. Lee, and S. Vaithyanathan (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques". In: *2002 Conference on Empirical Methods in Natural Language Processing 2002 (EMNLP 2002)*. ACL, pp. 79–86.

Pontiki, M., D. Galanis, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar (2015). "SemEval-2015 Task 12: Aspect based Sentiment Analysis". In: *9th International Workshop on Semantic Evaluation (SemEval 2015)*. ACL, pp. 486–495.

Pontiki, M., D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. Al-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. Zafra, and G. Eryigit (2016). "SemEval-2016 Task 5: Aspect Based Sentiment Analysis". In: *10th International Workshop on Semantic Evaluation (SemEval 2016)*. ACL, pp. 19–30.

Pontiki, M., D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar (2014). "SemEval-2014 task 4: Aspect based Sentiment analysis". In: *8th International Workshop on Semantic Evaluation (SemEval 2014)*. ACL, pp. 27–35.

Ribeiro, M., K. Grolinger, H. El Yamany, W. Higashino, and M. Capretz (2018). "Transfer Learning with Seasonal and Trend Adjustment for Cross-Building Energy Forecasting". In: *Energy and Buildings* 165, pp. 352–363.

Rietzler, A., S. Stabinger, P. Opitz, and S. Engl (2020). "Adapt or Get Left Behind: Domain Adaption through BERT Language Model Finetuning for Aspect-Target Sentiment Classification". In: *12th Conference on Language Resources and Evaluation Conference (LREC 2020)*. ELRA, pp. 4933–4941.

Schouten, K. and F. Frasincar (2016). "Survey on Aspect-Level Sentiment Analysis". In: *IEEE Transactions on Knowledge and Data Engineering*. LNCS 28 (3), pp. 813–880.

— (2018). "Ontology-Driven Sentiment Analysis of Product and Service Aspects". In: *15th International Conference of European Semantic Web (ESWC 2018)*. Vol. 10843. LNCS. Springer, pp. 608–623.

Taboada, M., J. Brooke, M. Tofiloski, K. Voll, and M. Stede (2011). "Lexicon-Based Methods for Sentiment Analysis". In: *Computational Linguistics* 37 (2), pp. 267–307.

Tankovska, H. (2021). *Social Media - Statistics and Facts*. URL: `https://www.statista.com/topics/1164/social-networks/`.

Thet, T., J. Na, and C. Khoo (2010). "Aspect-Based Sentiment Analysis of Movie Reviews on Discussion Boards". In: *Journal of Information Science* 36 (6), pp. 823–848.

Tian, Y., L. Yang, Y. Sun, and D. Liu (2021). "Cross-Domain End-to-End Aspect-Based Sentiment Analysis with Domain-Dependent Embeddings". In: *Complexity*, pp. 1–11.

Towell, G. and J. Shavlik (1994). "Knowledge-Based Artificial Neural Networks". In: *Artifical Intelligence* 70 (1-2), pp. 119–165.

Trusca, M., D. Wassenberg, F. Frasincar, and R. Dekker (2020). "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)*. Vol. 12128. LNCS. Springer, pp. 365–380.

Uzair, M. and N. Jamil (2020). "Effects of Hidden Layers on the Efficiency of Neural networks". In: *23rd International Multitopic Conference (INMIC 2020)*. IEEE.

Verhagen, W., P. Bermell-Garcia, R. Van Dijk, and R. Curran (2012). "A Critical Review of Knowledge-Based Engineering: An Identification of Research Challenges". In: *Advanced Engineering Informatics* 26 (1), pp. 5–15.

Wallaart, O. and F. Frasincar (2019). "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)*. Vol. 11503. LNCS. Springer, pp. 363–378.

Wang, F. and Q. Zhang (1997). "Knowledge-Based Neural Models for Microwave Design". In: *IEEE Transactions on Microwaves Theory and Techniques* 45 (12), pp. 2333–2343.

Wang, Z., M. Huang, L. Zhao, and X. Zhu (2016). "Attention-based LSTM for Aspect-level Sentiment Classification". In: *2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*. ACL, pp. 606–615.

Yanase, T., K. Yanai, M. Sato, T. Miyoshi, and Y. Niwa (2016). "bunji at SemEval-2016 Task 5: Neural and Synctactic Models of Entity-Attribute Relationship for Aspect-Based Sentiment Analysis". In: *10th International Workshop on Semantic Evaluation (SemEval 2016)*. ACL, pp. 289–295.

Yang, Z., D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy (2016). "Hierarchical Attention Networks for Document Classification". In: *2016 Conference of the North American Chaptor of the Association for Computational-Linguistics-Human Language Technologies (NAACL-HLT 2016)*. ACL, pp. 1480–1489.

Ying, W., K. Kann, M. Yu, and H. Schutze (2017). "Comparative Study of CNN and RNN for Natural Language Processing". In: arXiv preprint arXiv: 1702.01923.

Yosinski, J., J. Clune, Y. Bengio, and H. Lipson (2014). "How Transferable are Features in Deep Neural Networks?" In: *27th Annual Conference on Neural Information Processing Systems (NIPS 2014)*. Vol. 27. Curran Associates, pp. 3320–3328.

Yuan, J., Y. Zhao, B. Qin, and T. Liu (2021). "Learning to Share by Masking the Non-shared for Multi-domain Sentiment Classification". In: arXiv preprint arXiv: 2104.08480.

Zhang, H., I. Goodfellow, D. Metaxas, and A. Odena (2019). "Self-Attention Generative Adversarial Networks". In: *36th International Conference on Machine Learning (ICML 2019)*. Vol. 97. PMLR, pp. 7354–7363.

Zhang, W., W. Ouyang, W. Li, and D. Xu (2018). "Collaborative and Adversarial Network for Unsupervised Domain Adaption". In: *2018 Conference on Computer Vision and Pattern Recognition (CVPR 2018)*. IEEE, pp. 3801–3809.

Zhang, Y., Z. Qiu, T. Yao, D. Liu, and T. Mei (2018). "Fully Convolutional Adaption Networks for Semantic Segmentation". In: *2018 International Conference on Computer Vision and Pattern Recognition (CVPR 2018)*. IEEE, pp. 6810–6818.

Zhao, B., X. Li, and X. Lu (2019). "CAM-RNN: Co-Attention Model Based RNN for Video Captioning". In: *IEEE Transactions on Image Processing* 28 (11), pp. 5552–5565.

Zhao, J., T. Zhu, and M. Lan (2014). "ECNU: One Stone two birds: Ensemble of Heterogeneous Measures for Semantic Relatedness and Textual Entailment". In: *8th International Workshop on Semantic Evaluation (SemEval 2014)*. ACL.

Zheng, L., Y. Zhang, Y. Wu, Y. Wei, and Q. Yang (2017). "End-to-End Adversarial Memory Network for Cross-Domain Sentiment Classification". In: *26th International Joint Conference on Artificial Intelligence (IJCAI 2017)*. IJCAI, pp. 2237–2243.

Zheng, S. and R. Xia (2018). “Left-Center-Right Separated Neural Network for Aspect-based Sentiment Analysis with Rotatory Attention”. In:

Chapter 7

Appendix A: Descriptive statistics of data sets

The following tables provide the number of aspects per data set as well as the distribution of positive, neutral, and negative classified aspect sentiments.

Table 7.1 The distribution of the aspect sentiment polarities on the SemEval 2014 Restaurants domain

	Train	**Test**
Positive	60.1%	65.1%
Neutral	17.7%	17.5%
Negative	22.2%	17.4%
Total	3600	1122

Table 7.2 The distribution of the aspect sentiment polarities on the SemEval 2014 Laptops domain

	Train	**Test**
Positive	42.7%	52.4%
Neutral	19.8%	26.2%
Negative	37.5%	21.4%
Total	2250	701

Table 7.3 The distribution of the aspect sentiment polarities based on the Amazon/LibraryThing 2019 Books domain

	Train	**Test**
Positive	25.8%	32.3%
Neutral	63.1%	57.1%
Negative	11.1%	10.6%
Total	2700	804

Chapter 8

Appendix B: Detailed results for lambda optimisation

Table 8.1 shows the influence of λ on the performance of the DAT-LCR-Rot-hop++. First, the test accuracy represents the accuracy of the class discriminator for the test set of the target domain. The train accuracy equals the accuracy of the class discriminator on the training set of the source domain. Furthermore, the maximum accuracy is the highest testing accuracy achieved during all epochs. Last, the positive, neutral, and negative accuracies represent the accuracy of the class discriminator towards these sentiments on the test set of the target domain. These three scores combined, equal the test accuracy of the first column.

Table 8.1 The labeling accuracies of the six domain combinations for different values of λ

	λ	**test acc**	**train acc**	**max acc**	**positive acc**	**neutral acc**	**negative acc**
	0.5	67%	76%	70%	91%	11%	75%
	0.6	65%	75%	68%	86%	7%	85%
	0.7	65%	76%	67%	89%	16%	68%
rest-lapt	0.8	64%	77%	69%	85%	7%	82%
	0.9	65%	76%	68%	86%	4%	88%
	1.0	64%	76%	68%	83%	9%	85%
	1.1	68%	77%	68%	89%	19%	77%
	0.5	35%	82%	50%	87%	3%	46%
	0.6	34%	79%	56%	97%	1%	22%
	0.7	33%	81%	61%	98%	0%	15%
	0.8	33%	81%	48%	75%	2%	71%
rest-book	0.9	36%	80%	50%	93%	5%	29%
	1.0	45%	81%	51%	74%	26%	56%
	1.1	36%	81%	50%	99%	5%	2%
	0.5	68%	81%	74%	79%	27%	71%
	0.6	71%	82%	72%	80%	29%	76%
	0.7	68%	81%	73%	80%	0%	90%
lapt-rest	0.8	68%	80%	73%	81%	5%	83%
	0.9	62%	81%	73%	69%	5%	93%
	1.0	70%	80%	73%	86%	5%	77%
	1.1	72%	82%	73%	92%	0%	71%
	0.5	32%	73%	36%	58%	0%	92%
	0.6	35%	74%	38%	76%	1%	87%
	0.7	30%	74%	35%	63%	1%	91%
lapt-book	0.8	34%	75%	40%	69%	4%	89%
	0.9	38%	73%	38%	88%	4%	68%
	1.0	31%	74%	36%	65%	0%	92%
	1.1	33%	72%	35%	70%	1%	91%
	0.5	41%	83%	66%	46%	61%	2%
	0.6	58%	78%	71%	69%	66%	8%
	0.7	64%	77%	67%	75%	78%	10%
book-rest	0.8	63%	84%	64%	83%	48%	0%
	0.9	42%	84%	54%	43%	64%	1%
	1.0	45%	83%	64%	45%	90%	1%
	1.1	64%	82%	66%	86%	48%	1%
	0.5	52%	84%	62%	44%	97%	0%
	0.6	59%	80%	61%	88%	49%	0%
	0.7	49%	84%	62%	50%	88%	0%
book-lapt	0.8	54%	83%	60%	60%	89%	0%
	0.9	53%	82%	61%	62%	79%	0%
	1.0	57%	83%	59%	77%	64%	0%
	1.1	57%	83%	60%	6%	100%	0%

Chapter 9

Code DAT-LCR-Rot-hop++

Cross-Domain (CD) Aspect-Based Sentiment Classification (ABSC) using LCR-Rot-hop++ with Domain Adversarial Training (DAT).

9.1 Set-up instructions.

- Make sure you have a recent release of Python installed (we used Python 3.7), if not download from:
- Make sure you have a recent release of Python installed (we used Python 3.7), if not download from: https://www.python.org/downloads/
- Download Anaconda: https://www.anaconda.com/products/individual
- Set-up a virtual environment in Anaconda using Python 3.5 in order to be able to download tensorflow 1.5 package. Newer versions of python are not compatible.
- Copy all software from this repository into a file in the virtual environment.
- Open your new environment in the command window ('Open Terminal' in Anaconda)
- Navigate to the file containing all repository code (file_path) by running: cd file_path
- Install the requirements by running the following command: pip install -r requirements.txt
- You can open and edit the code in any editor, we used the PyCharm IDE: https://www.jetbrains.com/pycharm/

9.2 How to use?

- Make sure the Python interpreter is set to your Python 3.5 virtual environment (we used PyCharm IDE).
- Adjust the paths in config.py, main_test.py, and main_hyper.py

- Get raw data for your required domains by running raw_data.py for restaurant, laptop,and book domain.
- Get BERT embeddings by running files in getBERT for your required domains using Google Colab to obtain BERT embeddings (see files for further instructions on how to run).
- Prepare BERT train and test file and BERT embedding by running prepare_bert.py for your required domains.
- Tune hyperparameters to your specific task using main_hyper.py or use hyperparameters as pre-set in main_test.py.
- Select tests to run and run main_test.py (running all tests will take a long time, 4-5 minutes per epoch).
- Make sure write_result is set to True if you want the results to be saved to a text file.

9.3 References

This code is adapted from Trusca, Wassenberg, Frasincar and Dekker (2020).
https://github.com/mtrusca/HAABSA_PLUS_PLUS
Truşcă M.M., Wassenberg D., Frasincar F., Dekker R. (2020) 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.