Fine-Tuning for Cross-Domain Aspect-Based Sentiment Classification

Stefan van Berkum
Erasmus University Rotterdam
Rotterdam, the Netherlands
stefanvanberkum@gmail.com

Sophia van Megen
Erasmus University Rotterdam
Rotterdam, the Netherlands
sophiavanmeg@gmail.com

Max Savelkoul
Erasmus University Rotterdam
Rotterdam, the Netherlands
mvf.savelkoul@gmail.com

Pim Weterman
Erasmus University Rotterdam
Rotterdam, the Netherlands
wetermanpim@gmail.com

Flavius Frasincar
Erasmus University Rotterdam
Rotterdam, the Netherlands
frasincar@ese.eur.nl

ABSTRACT

Aspect-Based Sentiment Classification (ABSC) is a subfield of sentiment analysis concerned with classifying sentiment attributed to pre-identified aspects. A problem in ABSC nowadays is the limited availability of labeled data for certain domains. This study aims to improve sentiment classification accuracy for these domains where labeled data is scarce. Our proposed approach is to apply cross-domain fine-tuning to a state-of-the-art deep learning method designed for ABSC: LCR-Rot-hop++. For this purpose, we initially train the model on a domain that has a lot of labeled data available and consecutively fine-tune the upper layers with training data of the target domain. The performance of the fine-tuning method is evaluated relative to a model that is trained from scratch for each target domain. For the initial training, restaurant review data is used. For the fine-tuning and from-scratch training we use review data for laptops, books, hotels, and electronics. Our results show that when comparing the fine-tuning with the from-scratch method (for the same training set), the fine-tuning method on average outperforms the from-scratch method when the training set is small for *all* considered domains and is considerably faster.

CCS CONCEPTS

• **Information systems** → **Sentiment analysis**; *Information extraction*; Web mining.

ACM Reference Format:
Stefan van Berkum, Sophia van Megen, Max Savelkoul, Pim Weterman, and Flavius Frasincar. 2021. Fine-Tuning for Cross-Domain Aspect-Based Sentiment Classification. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI-IAT '21), December 14–17, 2021, ESSENDON, VIC, Australia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3486622.3494003

WI-IAT '21, December 14–17, 2021, ESSENDON, VIC, Australia

ACM ISBN 978-1-4503-9115-3/21/12…$15.00
https://doi.org/10.1145/3486622.3494003

1 INTRODUCTION

The digital age has brought ever-growing amounts of information to our fingertips. Thanks to the desire for sharing impressions and ideas, the Web is now evolving to be a forum for consumers to evaluate services and products based on feedback from other like-minded people. Consumers are not the only ones that benefit from this mutual content sharing, this user-generated Web-based content is also getting increasing attention as a source of data for its cross-domain business applications.

The increased amount of user-generated data gave rise to another problem, people now face an excess of opinionated texts that have to be filtered to get the desired information. This spawned interest in automatic approaches that provide summaries of people's sentiments. One of these approaches is sentiment analysis. This method's main purpose is to detect sentiment and opinions from text, and combine this information into useful and quickly interpretable results for businesses and consumers. Sentiment analysis can be a useful technique when the quantity of data is too large for companies to process manually.

A subfield of sentiment analysis is Aspect-Based Sentiment Analysis (ABSA) [13], which involves two steps. First, the aspects of a certain target object are identified, which is called Aspect Determination (AD). Secondly, the sentiment polarity, i.e., positive, negative, or neutral, with respect to the previously found aspects is determined, which is called Aspect-Based Sentiment Classification (ABSC). In this work, we will focus on ABSC.

Some recent methods used in ABSC are hybrid methods [3]. These methods make use of a combination of ontology reasoning followed up by a neural network. [17] uses this approach in the Ont+LCR-Rot-hop method. We use the LCR-Rot-hop++ hybrid method discussed in [15] because this method showed superior performance to LCR-Rot-hop and solutions based on non-hybrid approaches (i.e., either ontology reasoning or machine learning).

A problem in ABSC nowadays is the limited availability of labeled data in certain domains. To combat this, we will focus on the cross-domain application of ABSC. We aim to explore how to adapt the neural network such that it works on other domains than it was initially trained for. This is a subfield of transfer learning called domain adaptation. In this work, we investigate how we can adapt a state-of-the-art deep learning solution for cross-domain ABSC.

Specifically, we initially train the entire LCR-Rot-hop++ model, as proposed in [15], on a domain where much labeled data is available. Based on the results of diagnostic classification for LCR-Rot-hop [10], we proceed by freezing the weights of the lower layers of the model, i.e., the Left-Center-Right (LCR) Bi-directional Long-Short-Term-Memory (Bi-LSTM) modules, and fine-tune the remaining layers on the target domain. For this last step we take multiple subsets of the available data, to test how our approach performs when there is only limited labeled data available for the target domain.

To test our method we use two benchmark methods. First, we train LCR-Rot-hop++ on the target domain and subsequently test it on that same domain. This allows us to compare performance of fine-tuning to that of from-scratch training. As a lower bound, we train LCR-Rot-hop++ on a different domain than the target domain and simply test it on the target domain.

We expect the fine-tuning method to perform relatively well for small subsets of target domain data. As the size of the training set increases, however, we expect the accuracy gain from fine-tuning to shrink up to a point where eventually the from-scratch method outperforms the fine-tuning (for large training sets). This hypothesis is based on the fact that the lower layers that are frozen in fine-tuning mainly encode general language characteristics. Therefore, we would expect this pattern to occur because the lower layers might still capture some (but not a lot of) domain-specific information. A major benefit of freezing certain layers during fine-tuning is that it can substantially speed up model training, which is useful if we want to quickly adapt our model to other domains.

All source code and data used in this paper can be found at https://github.com/stefanvanberkum/CD-ABSC. The remainder of this paper is structured as follows. Section 2 discusses the academic literature related to our research. Section 3 briefly discusses the data used in this paper. Section 4 briefly outlines the original LCR-Rot-hop++ and describes our fine-tuning extension as well as how its performance is evaluated. Section 5 reports the obtained results and discusses their interpretation. Last, Sect. 6 summarizes the results and discusses some suggestions for future work.

2 RELATED WORK

Sentiment polarity analysis is a natural language processing technique used to detect positive, negative, or neutral sentiments in subjective information. The method allows for automatic analysis of customer feedback. We can distinguish sentiments on the word, sentence, or document level. Because of the ever-rising amount of reviews on the Web, there is a high demand for machine learning methods that correctly and efficiently classify sentiments.

2.1 ABSC

In our paper, we focus on a subfield of sentiment analysis, namely ABSC. In ABSC, a sentiment towards a particular aspect or feature is determined. For example, in the sentence "the display of the laptop is of poor quality", a negative sentiment towards the display of the laptop is expressed, not directly towards the laptop itself. Hence, display is one aspect of sentence-level sentiment.

In [13] an in-depth overview of ABSC is given, in addition to recent progress in the field. Two distinctive methods that are mentioned are the knowledge-based approach and the machine learning-based approach. In the former, domain-specific information is used to determine which words express sentiment. The ontology-based technique proposed in [14] has a competitive performance of over 80% accuracy for SemEval 2015 and SemEval 2016 data. Other knowledge-based methods for sentiment classification are based on Part-of-Speech (PoS) tagging and domain vocabulary [7]. Machine learning methods also achieved success in ABSC recently. Using advanced representation ability, neural networks can automatically generate meaningful representations for the aspects and their contexts, and obtained outstanding results. In our research, we will use one of such methods proposed in [15], which showed results superior to many state-of-the-art solutions when paired with ontology reasoning. This method (Ont+LCR-Rot-hop++) is an extension of the Ont+LCR-Rot-hop as proposed in [17], which is in turn an extension of the LCR separated neural network with Rotatory attention (LCR-Rot) approach given in [18]. As our proposed method mainly pertains to the machine learning part of the Ont+LCR-Rot-hop++, we will focus on just the LCR-Rot-hop++ part in our analysis (i.e., without the ontology).

2.2 Cross-Domain Sentiment Classification

Cross-domain sentiment classification is a hot topic in the sentiment analysis field of research. The goal is to generalize a classifier that is trained on the source domain to a different target domain. Acquiring labeled training data is a costly process, especially for sentiment classification, which makes it difficult to train machine learning models for new tasks. By using cross-domain methods, less costly labeled training data is needed to develop new machine learning tasks.

An approach proposed in [5] is Domain Adversarial Neural Networks (DANN), this model trains on features that are domain-invariant using adversarial training. The goal is to embed domain adaptation in the process of deep learning, so that the final sentiment classification decision is made based on features that are both discriminative and domain-invariant. The proposed architecture is a standard feed-forward neural network with one additional layer, the gradient reversal layer. This last layer reverses the gradient (maximizes the loss) for domain classification during backpropagation, such that the domain classifier is not able to predict the domain of the encoded representation.

Another approach that uses adversarial training is an end-to-end Adversarial Memory Network (AMN) [9]. Its goal is to improve the interpretability of deep neural networks. Similarly to [5], the authors use a framework where they jointly train two classifiers, one for sentiment classification and one for domain classification. The proposed AMN is able to capture the pivots, which are words important for sentiment classification and shared in both domains, by using an attention mechanism.

Yet another solution for cross-domain sentiment classification is to mask domain specific words to train the neural network on domain-invariant features. This type of pre-training is also incorporated into the language representation model BERT, as proposed in [4].

Other methods focus on freezing the weights of the lower layers of the neural network and fine-tuning the rest of the model using labeled information from the target domain. [8] freezes the first few layers of pre-trained language models BERT and RoBERTa, and fine-tunes the other layers. The authors find that only a fourth of the final layers need to be fine-tuned to achieve 90% of the original accuracy. In our paper we will use a similar freezing approach applied to the neural network part of Ont+LCR-Rot-hop++ in [15], as this method shows superior performance to Ont+LCR-Rot-hop and solutions based on non-hybrid approaches.

2.3 Diagnostics

Many state-of-the-art ABSC methods produce so-called black box models, where the sentiments towards a certain aspect are difficult to interpret. [10] provides an explanation of modern neural attention models, using a technique called diagnostic classification. The paper proposes explanation models for the LCR-Rot-hop model proposed in [17]. The authors conclude that lower parts of the LCR-Rot-hop model encode sentiment value and PoS, whereas the upper layers determine the presence of a relation with the aspect and the sentiment value of words related to the aspect.

Taking this into account, we opt to freeze the weights and biases of the lower parts of LCR-Rot-hop++ in our proposed fine-tuning approach, namely the LCR Bi-LSTMs. We only fine-tune the model with target-domain data for the remaining layers.

3 DATA

In this paper, we use five different datasets for a total of seven different domains. We use the Semantic Evaluation (SemEval) 2014 dataset [12] for the restaurant and laptop domain, and SemEval 2015 dataset [11] for the hotel domain. For the book domain we use the Amazon/LibraryThing 2018 dataset [1] and for the electronics domain the Customer Reviews Dataset (CRD) 2004 [6]. SemEval datasets are widely used in research papers concerning the subject of cross-domain sentiment analysis [12, 15, 17]. These particular datasets are convenient since we can easily compare our findings with the findings in related literature. We use the SemEval 2014 restaurant review data to initially train the LCR-Rot-hop++ neural network by [15] for subsequent fine-tuning, this domain is discussed in Subsect. 3.1. For every other (target) domain, we take a given number of training subsets of increasing size, where aspect-based opinions (sentences that contain sentiment regarding a particular aspect) are cumulatively included. In other words, each training set consists of the previous (smaller) training set and the new set of opinions. These target domains are discussed in Subsect. 3.2.

3.1 Original Domain

The data used to train the neural network for subsequent fine-tuning consists of restaurant reviews from the SemEval 2014 dataset [12]. Every review sentence contains one or more opinions. In the sentences, an opinion about a certain aspect of the restaurant is given, and these aspects are divided into aspect categories. It is important to note that in this work the aspects are provided in each dataset. For each given aspect we aim to predict the polarity of the sentiment associated with it in the review sentence. The polarity regarding the aspects can either be positive, neutral, or negative.

```
<sentence id="1206">
    <text>The place is small and cramped but the food
         is fantastic.
    </text>
    <aspectTerms>
        <aspectTerm term="place" polarity="negative"
            from="4" to="9"/>
        <aspectTerm term="food" polarity="positive"
            from="39" to="43"/>
    </aspectTerms>
    <aspectCategories>
        <aspectCategory category="ambience" polarity="
            negative"/>
        <aspectCategory category="food" polarity="
            positive"/>
    </aspectCategories>
</sentence>
```

Figure 1: Example of sentence in a restaurant review in the XML markup language.

The review sentences for this dataset are presented in the XML markup language. Figure 1 shows an example of such a sentence. This particular sentence consists of two opinions. For each opinion, a polarity is included which indicates whether the writer of the review is positive, neutral, or negative towards that particular aspect. The sentence in Fig. 1 illustrates that it is possible to have more than one opinion in a sentence. In case that we have more than one sentiment type (e.g., positive and negative) associated to the very same aspect, we are dealing with a conflicting sentiment. A sentence could also contain implicit aspects. In this work, we do not consider conflicting sentiment and implicit aspects, and remove all such opinions from the dataset, because the machine learning model is not able to cope with such cases. For the restaurant training dataset, we have 3693 opinions. 2.5% of the opinions have a conflicting sentiment and are therefore removed. No opinions are removed because of implicit aspects in this dataset.

Below the sentence in Fig. 1, the aspects and the corresponding polarities in that sentence are identified. In this sentence the aspects are "place" and "food", and they have a negative and positive polarity, respectively. Next to the polarity, the position of the aspect in the sentence is given, but these are not considered in this work. The last few lines in the XML code give the aspect category of the aspects in the sentence and their corresponding polarities towards that aspect category, which are also not considered in this work.

3.2 Target Domains

In this section we describe the datasets of the target domains that we consider in this work: laptops, books, hotels, and electronics. These sets are all split into a train and test set using approximately 80% and 20% of the opinions, respectively. The first domain that is considered during the analysis is laptops. This set can be split into nine training subsets that each increase in size by 250 opinions. That is, the first subset contains the first 250 opinions, and each

of the following subsets contains the previous subset and the next 250 opinions. For consistency, the other large dataset (books) is also split into nine training subsets. Each of the smaller datasets (hotels and electronics) is, however, split into ten training subsets for convenient data representation.

3.2.1 Laptops. The laptop reviews data is also from the SemEval 2014 dataset [12] and is structured in a similar way as the restaurant data. We vary the amount of opinions from the laptop review dataset used to fine-tune the neural network. The 2250 opinions used for training are split into nine training subsets of increasing size. The test set consists of 701 opinions. 2.0% of the opinions in the laptop dataset have a conflicting sentiment and are therefore removed. No opinions in this domain are removed because of implicit aspects.

3.2.2 Books. Another domain used for fine-tuning and to test the cross-domain performance is the book domain. The book review data is from Amazon/LibraryThing 2018 [1]. The 2700 opinions used for training are again split in nine training subsets of increasing size. The test set consists of 804 opinions. No opinions are removed from the dataset because of conflicting sentiments. 8.6% of the total opinions in this dataset are removed because of implicit aspects.

3.2.3 Hotels. Yet another dataset used to fine-tune our model consists of hotel reviews and is from SemEval 2015 [11]. The hotel training set only contains 200 opinions and is split into ten training subsets of increasing size. The test set consists of 64 opinions. No conflicting sentiments are present in the hotel dataset. However, 22.1% of the total opinions in this dataset are removed because of implicit aspects.

3.2.4 Electronics. The electronics domain consists of four subdomains, i.e., DVD players, digital cameras, MP3 players, and cell phones, and is from CRD 2004, which was first introduced in [6]. CRD 2004 contains two separate datasets for digital camera reviews, which we combine to have a slightly larger dataset. The training set for each of these domains is again split into ten training subsets of increasing size. For the electronics dataset, the sentiment towards a particular aspect can only be positive or negative, not neutral like the other domains. The dataset for electronics assigns a number to each opinion indicating its strength, where (-)3 is the strongest and (-)1 the weakest. We set all these numbers equal to (-)1 as we do not differentiate between opinion strengths in this work. Furthermore, no opinions are removed because of conflicting sentiments for any subdomain. The percentages of total opinions removed because of implicit aspects for the DVD players, digital cameras, MP3 players, and cell phones are 27.4%, 19.2%, 20.3%, and 15.9%, respectively.

4 METHODOLOGY

Our proposed approach to cross-domain ABSC is to apply domain fine-tuning to the upper layers of the LCR-Rot-hop++ model described in [15]. First, the LCR-Rot-hop++ is briefly introduced in Subsect. 4.1. Our proposed approach to fine-tune the model for cross-domain ABSC is discussed in Subsect. 4.2. Lastly, the evaluation of the fine-tuning approach will be briefly outlined in Subsect. 4.3.

4.1 LCR-Rot-hop++

The LCR-Rot-hop++ method [15] uses three Bi-LSTM modules, a rotary, and a hierarchical attention mechanism to obtain sentiment classifications. In Fig. 2, we visualize this method. The solid (red) arrows depict the first step, which is the layer of three Bi-LSTMs. The densely dashed (yellow) arrows show the rotary attention mechanism with multiple hops. The en dashed (green) arrows depict the hierarchical attention mechanism. Finally, the em dashed (blue) arrows depict the representation concatenation and sentiment classification. This subsection is meant to provide a brief overview of the LCR-Rot-hop++ method.

The model's input is defined as a sentence $S = [w_1, w_2, ..., w_N]$ with N words, where w_i denotes the i^{th} word of the sentence. This sentence S is split into three parts: left context $[w_1^l, w_2^l, ..., w_L^l]$, target (aspect) $[w_1^t, w_2^t, ..., w_T^t]$, and right context $[w_1^r, w_2^r, ..., w_R^r]$, where L, T, and R denote the lengths of the three parts, respectively. For example, if we have input sentence "the food was great", then this is split into: ["the"] (left context), ["food"] (target), and ["was", "great"] (right context). The words in the sentence are then embedded using the pre-trained BERT-Base model with 12 layers and a hidden layer size of 768 [16].

The next layer in the model consists of three Bi-LSTMs. Each of these three Bi-LSTM modules deals with a different part of the input sentence. More specifically, the model has a target (center) Bi-LSTM that deals with the target phrase and a left and right Bi-LSTM that deal with the left and right context, respectively. The output of these Bi-LSTMs is called a hidden state vector and this is fed to the rotary attention mechanism.

The rotary attention mechanism first uses a pooling operation to obtain an average representation of the target phrase, which is then used to calculate the representation of the left and right context phrases using a bilinear attention mechanism. These representations are weighted by applying a hierarchical attention mechanism. Consecutively, these attention-weighted left and right context representations are used to obtain a left- and right-aware representation of the target phrase, respectively, again using a bilinear attention mechanism. As before, these representations are weighted by applying another hierarchical attention mechanism. These can then be fed into the start of the rotary attention mechanism once more, instead of the output of the initial pooling operation. As in [15], this mechanism is repeated for three hops.

After executing all hops over the rotary attention mechanism, the final representations are again weighted by applying a hierarchical attention mechanism. After this last weighting, the final representations are concatenated and fed into a final Multi-Layer Perceptron (MLP) which uses a softmax function to compute aspect-level sentiment predictions (p). The model is trained with backpropagation and stochastic gradient descent optimization with momentum. To avoid overfitting, a dropout strategy that randomly removes nodes along with their incoming and outgoing connections from the network is used.

4.2 Domain Fine-tuning

Our proposed adaptation of the LCR-Rot-hop++ method to improve its cross-domain performance is fine-tuning of the model by training only the upper layers of the model with data from the target

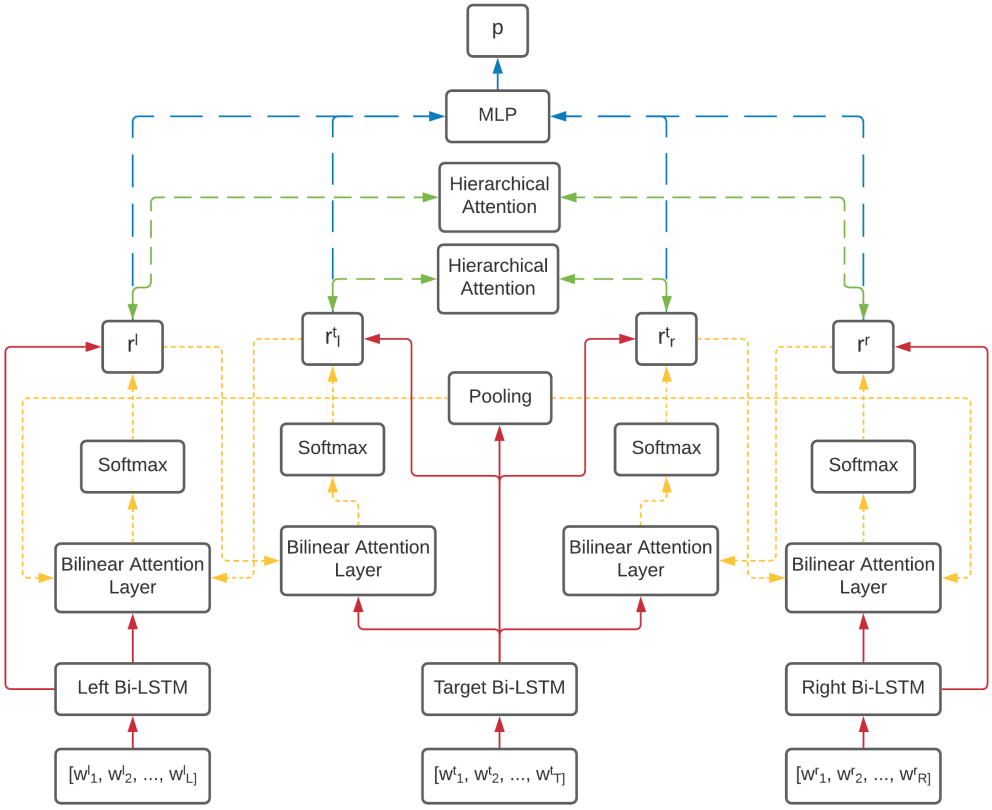


Figure 2: Visual representation of the LCR-Rot-hop++ method.

domain. The choice for which layers get fixed during fine-tuning is based on previous work on explaining which type of information gets encoded in each layer of LCR-Rot-hop [10]. The authors argue that the lower layers of the LCR-Rot-hop model mainly encode general language characteristics, while the upper layers represent the more domain-specific language and associated sentiment. The LCR-Rot-hop++ is very similar to the LCR-Rot-hop method, the only differences being the use of BERT embeddings (as opposed to GloVe) and the use of hierarchical attention layers in the upper part of the model. Therefore, we assume that the Bi-LSTMs also mainly encode general-language information in the LCR-Rot-hop++ model. Consequently, in our proposed approach the weights for these Bi-LSTMs are fixed and we only fine-tune the model for the multi-hop rotary attention module with hierarchical attention and MLP, by training it on labeled data of the target domain. The fixed layers are depicted in Fig. 2 by the solid (red) arrows.

4.3 Evaluation

For this research, LCR-Rot-hop++ is initially trained on restaurant data. Consecutively, it is fine-tuned and tested on a different target domain (laptop, book, hotel, or electronics data). Two benchmark approaches can be used to evaluate the performance of the proposed cross-domain adaptation of the LCR-Rot-hop++: a restaurant-target and a target-target approach. In the restaurant-target approach, the testing data for each target is simply evaluated using a model trained only on restaurant data. The performance of this benchmark can be seen as a worst-case performance. The target-target approach uses the target training data meant for fine-tuning to train the model from scratch (without fixing any layers). These three approaches (i.e., restaurant-target, target-target, and fine-tuning) are evaluated for varying sizes of target training data. This implies that in relation to the size of the training set, the restaurant-target measure is the only one that has a constant value as it is not trained nor fine-tuned using target domain data.

For consistency, the same number of nodes is used for each layer as in [15], i.e., 300 nodes for each Bi-LSTM, 600 nodes for each bilinear and hierarchical attention layer and 2400 nodes for the MLP. Additionally, as in [15], the model is run for 200 iterations. Moreover, the hyperparameters (learning rate, dropout rate, momentum term, and L2 regularization term) are determined separately for each of the target domains for each of the three subtasks (i.e., restaurant-target, target-target, and fine-tuning). Here, the hyperparameters for the restaurant-target subtask are only tuned once, for the initial restaurant training. For this purpose a Tree-structured Parzen Estimator (TPE) is used [2]. Ideally, one would tune the hyperparameters for each different training set, for as many times as possible, and for as many iterations as in the regular runs (200). However, due to time-constraints we have to limit the hyperparameter tuning. Therefore, for each target domain and subtask, the algorithm is run ten times for 15 iterations of the model.

For the applications that use target training data (i.e., target-target and fine-tuning), the optimal hyperparameters are only determined for the largest target training set. The assumption is that limiting the hyperparameter tuning affects the accuracy for each subtask (especially target-target and fine-tuning) within a given target domain in a similar way, and that it will therefore have little effect on the ultimate difference in accuracies between these subtasks.

5 RESULTS

In this section we report the results of domain fine-tuning the LCR-Rot-hop++ model for the following domains: laptops, books, hotels, and electronics. For every domain, the accuracies obtained from fine-tuning on the target domain are compared with the two benchmark performance measures (restaurant-target and target-target) to evaluate the cross-domain adaptation of the LCR-Rot-hop++ method. For each of the figures in the following subsections, an OLS regression is displayed with a corresponding 95% confidence interval to approximate a trend for the target-target and fine-tuning results. This allows for an easier interpretation of the results and more importantly, it allows us to easily identify common tendencies between all different target domains.

5.1 Laptops

The laptop domain data is divided into nine subsets that cumulatively increase by 250 opinions. Figure 3 depicts the obtained accuracies for all training subsets.

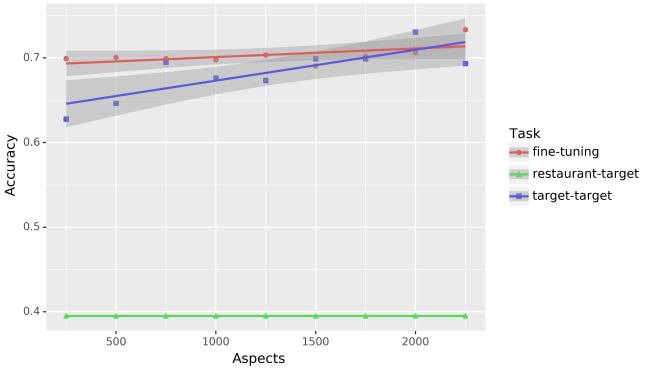


Figure 3: Test accuracy of restaurant training (restaurant-target), laptop training (target-target), and restaurant training with laptop fine-tuning (fine-tuning) for different laptop training set sizes.

The restaurant-target benchmark has an accuracy of approximately 40%. The target-target benchmark accuracy increases from approximately 65% to 72% as the amount of opinions used for training increases from 250 to 2250. Fine-tuning using the target domain gives an accuracy that increases from approximately 69% to 71% on the same training sets. Comparing the fine-tuning with the target-target benchmark, we see that the fine-tuning approach on average outperforms the target-target approach when the training set is small, with an expected intersection around 2059 opinions.

5.2 Books

The book domain data is divided into nine subsets that cumulatively increase by 300 opinions. Figure 4 depicts the obtained accuracies for all training subsets.

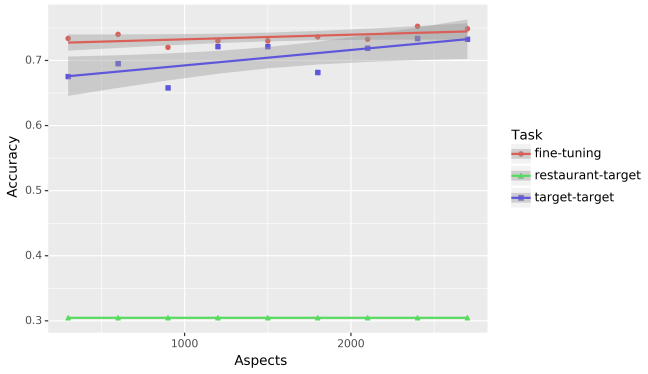


Figure 4: Test accuracy of restaurant training (restaurant-target), book training (target-target), and restaurant training with book fine-tuning (fine-tuning) for different book training set sizes.

The restaurant-target benchmark has an accuracy of approximately 30%. The target-target benchmark accuracy increases from approximately 68% to 73% as the amount of opinions used for training increases from 300 to 2700. Fine-tuning using the target domain gives an accuracy that increases from approximately 73% to 74% on the same training sets. Comparing the fine-tuning with the target-target benchmark, we see that the fine-tuning approach on average outperforms the target-target approach for all considered sizes of the training set, with an expected intersection around 3417 opinions.

5.3 Hotels

The hotel domain data is divided into ten subsets that cumulatively increase by 20 opinions. Figure 5 depicts the obtained accuracies for all training subsets.

The restaurant-target benchmark has an accuracy of approximately 47%. The target-target benchmark accuracy increases from approximately 60% to 67% as the amount of opinions used for training increases from 20 to 200. Fine-tuning using the target domain gives an accuracy that increases from approximately 69% to 74% on the same training sets. Comparing the fine-tuning with the target-target benchmark, we see that the fine-tuning approach on average outperforms the target-target approach for all considered sizes of the training set, with an expected intersection around 1289 opinions.

5.4 Electronics

The DVD player, digital camera, MP3 player, and cell phone domain datasets are divided into ten subsets that cumulatively increase by 25, 31, 54, and 22 opinions, respectively. Figure 6 depicts the obtained accuracies for all training subsets for each domain.

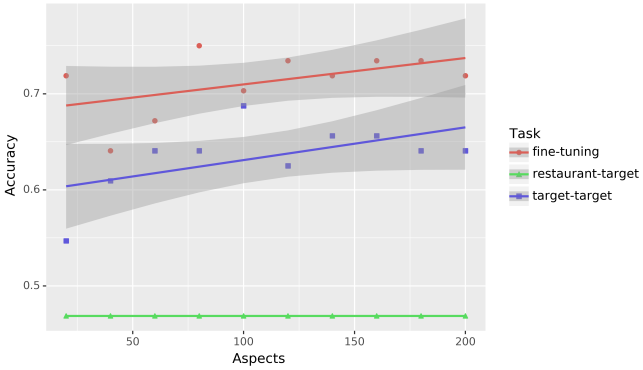


Figure 5: Test accuracy of restaurant training (restaurant-target), hotel training (target-target), and restaurant training with hotel fine-tuning (fine-tuning) for different hotel training set sizes.

For the DVD player domain depicted in Fig. 6a, the restaurant-target accuracy is approximately 49%. The target-target and fine-tuning accuracies increase from approximately 66% to 83% and 78% to 79%, respectively, with an expected intersection around 197 opinions.

For the camera domain depicted in Fig. 6b, the restaurant-target accuracy is approximately 59%. The target-target and fine-tuning accuracies increase from approximately 87% to 92% and 90% to 91%, respectively, with an expected intersection around 214 opinions.

For the MP3 player domain depicted in Fig. 6c, the restaurant-target accuracy is approximately 46%. The target-target and fine-tuning accuracies increase from approximately 66% to 80% and 75% to 79%, respectively, with an expected intersection around 487 opinions.

For the cell phone domain depicted in Fig. 6d, the restaurant-target accuracy is approximately 67%. The target-target and fine-tuning accuracies increase from approximately 72% to 93% and 79% to 88%, respectively, with an expected intersection around 151 opinions.

5.5 Discussion

In summary, it appears that our proposed fine-tuning approach initially outperforms the from-scratch training (target-target) for all considered domains. Individually, the regressions for each target domain might not be very informative as each one is only based on nine or ten data points. However, when we consider all graphs together, we can clearly see that there is a common tendency for all different target domains. That is, in each graph, the fine-tuning approach initially outperforms the target-target approach and the target-target approach overtakes the fine-tuning approach at the end of the graph. At the start of the graphs (the smallest training set), the fine-tuning approach outperforms the target-target approach by approximately 5 percentage points for each of the large datasets (laptops and books), and by approximately 2-11 percentage points (8 percentage points on average) for the small datasets (hotels and electronics). Because this common tendency holds for *all* seven domains, the results suggest that our proposed fine-tuning approach improves the obtained accuracy when only a limited amount of training data is available.

This tendency for the fine-tuning approach to outperform the target-target approach for small training sets and for the target-target approach to perform better near the end is also what we would expect, theoretically, given that we fix certain layers during fine-tuning. As mentioned in Subsect. 4.2, the layers that are fixed during fine-tuning mainly encode general language characteristics, while the layers that are not fixed encode the more domain-specific language and associated sentiment. For this reason, it is understandable that a model that is pre-trained on reviews for another domain (restaurants in our case) and consecutively fine-tuned with a small amount of target domain training data, can outperform a model that has to be trained from-scratch (target-target) with that same small amount of data. When the size of the target domain training set is large, however, we would expect the from-scratch training to outperform the fine-tuning approach as the layers that are fixed during fine-tuning might still capture some domain-specific information.

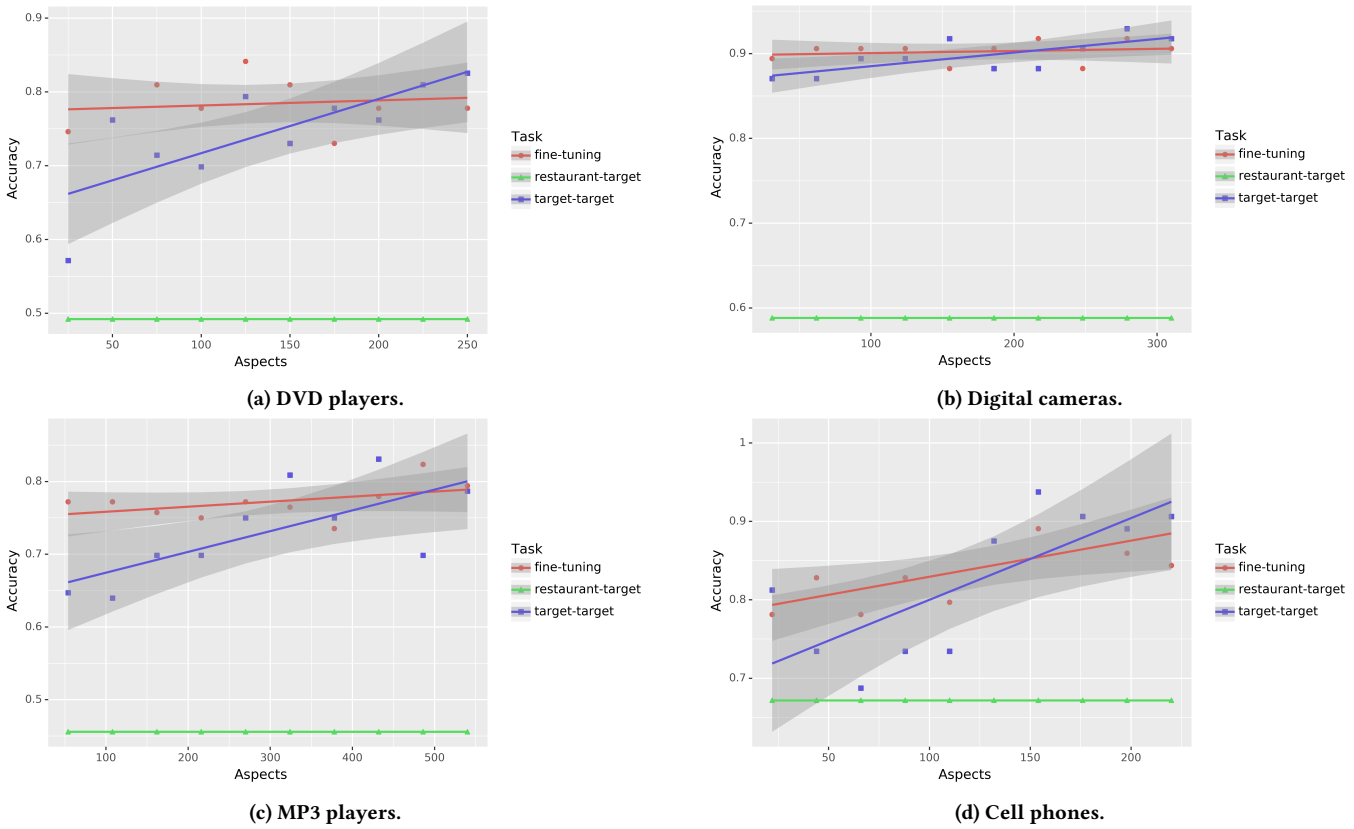
The advantage of applying this particular fine-tuning technique to the LCR-Rot-hop++ method, is that we can exploit its architecture to freeze layers that are expected to encode similar information across domains. This can substantially speed up model training. To illustrate this, our proposed fine-tuning approach is approximately twice as fast per iteration for small amounts of training data, compared to training the full model with that same amount of data (i.e., from-scratch training). This difference in speed only increases when the model is trained using larger amounts of training data.

6 CONCLUSION

The amount of opinionated data on the Web is extensive. However, annotated data is scarce for certain domains and acquiring these is a labor intensive task. Therefore, in this paper we propose an adaptation of a state-of-the-art ABSC method to perform cross-domain analysis. Specifically, we train the the LCR-Rot-hop++ model [15] on restaurant data. We then freeze lower layers (the LCR Bi-LSTMs), and train the remaining layers on the target domain. A major benefit of this approach is that it can substantially speed up model training.

From the obtained accuracies we can derive one common tendency, namely that our proposed fine-tuning approach on average outperforms the regular from-scratch training when the training set is small. Based on the large datasets (laptops and books), the accuracy gain from our proposed fine-tuning approach compared to the target-target approach is approximately 5 percentage points, when the target training set is small. For the small datasets (hotels and electronics), this accuracy gain is approximately 8 percentage points on average.

The choice to freeze the LCR Bi-LSTMs is based on the assumption that these layers encode mainly general language characteristics, while the upper layers encode more domain-specific language and associated sentiment. This assumption is based on a diagnostic classification analysis for LCR-Rot-hop [10]. Since we make use of the LCR-Rot-hop++ method [15] in this paper, suggestions for future research would include doing diagnostic classification analysis for LCR-Rot-hop++ to determine the validity of this assumption.



(a) DVD players. **(b) Digital cameras.**

(c) MP3 players. **(d) Cell phones.**

Figure 6: Test accuracy of restaurant training (restaurant-target), target training (target-target), and restaurant training with target fine-tuning (fine-tuning) for different electronics training set sizes.

REFERENCES

[1] Tamara Álvarez-López, Milagros Fernández-Gavilanes, Enrique Costa-Montenegro, and Patrice 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) (LNCS, Vol. 10859)*. Springer, 3–14.

[2] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for Hyper-Parameter Optimization. In *25th Annual Conference on Neural Information Processing Systems (NIPS 2011)*, Vol. 24. Curran Associates, Inc., 2546–2554.

[3] Erik Cambria, Soujanya Poria, Alexander Gelbukh, and Mike Thelwall. 2017. Sentiment Analysis Is a Big Suitcase. *IEEE Intelligent Systems* 32, 6 (2017), 74–80.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2019)*. ACL, 4171–4186.

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-Adversarial Training of Neural Networks. In *Journal of Machine Learning Research*, Vol. 17. 59:1–59:35.

[6] Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *10th International Conference on Knowledge Discovery and Data Mining (KDD 2004)*. ACM, 168–177.

[7] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. In *8th International Workshop on Semantic Evaluation (SemEval 2014)*. ACL, 437–442.

[8] Jaejun Lee, Raphael Tang, and Jimmy Lin. 2019. What would Elsa do? Freezing layers during transformer fine-tuning. In *Computing Research Repository*. arXiv preprint arXiv:1911.03090.

[9] Zheng Li, Yu Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017. End-to-End Adversarial Memory Network for Cross-domain Sentiment Classification. In *Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017)*. IJCAI.org, 2237–2243.

[10] Lisa Meijer, Flavius Frasincar, and Maria Mihaela Truşcă. 2021. Explaining a Neural Attention Model for Aspect-Based Sentiment Classification Using Diagnostic Classification. In *36th ACM Symposium on Applied Computing (SAC 2021)*. ACM, 821–827.

[11] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In *9th International Workshop on Semantic Evaluation (SemEval 2015)*. ACL, 486–495.

[12] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *8th International Workshop on Semantic Evaluation (SemEval 2014)*. ACL, 27–35.

[13] Kim Schouten and Flavius Frasincar. 2016. Survey on Aspect-Level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering* 28, 3 (2016), 813–830.

[14] Kim Schouten and Flavius Frasincar. 2018. Ontology-Driven Sentiment Analysis of Product and Service Aspects, In 15th Extended Semantic Web Conference (ESWC 2018). *The Semantic Web Lecture Notes in Computer Science* 10843, 608–623.

[15] Maria Mihaela Truşcă, Daan Wassenberg, Flavius Frasincar, and Rommert 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) (LNCS, Vol. 12128)*. Springer, 365–380.

[16] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. *arXiv preprint arXiv:1908.08962* (2019).

[17] Olaf Wallaart and Flavius Frasincar. 2019. A Hybrid Approach for Aspect-Based Sentiment Analysis Using a Lexicalized Domain Ontology and Attentional Neural Models. In *16th Extended Semantic Web Conference (ESWC 2019) (LNCS, Vol. 11503)*. Springer, 363–378.

[18] Shiliang Zheng and Rui Xia. 2018. Left-Center-Right Separated Neural Network for Aspect-based Sentiment Analysis with Rotatory Attention. In *Computing Research Repository*. arXiv preprint arXiv:1802.00892. arXiv:1802.00892