Preventing Negative Transfer on Sentiment Analysis in Deep Transfer Learning

Osayande P. Omondiagbe[1,2,*], Sherlock A. Licorish[2] and Stephen G. MacDonell[2,3]

[1]Department of Informatics, Landcare Research, Lincoln, New Zelaand
[2]Department of Information science, University of Otago, Dunedin, New Zelaand
[3]Software Engineering Research Lab, Auckland University of Technology, Auckland, New Zealand

Abstract

Data sparsity is a challenge facing most modern recommendation systems. With cross-domain recommendation technique, one can overcome data sparsity by leveraging knowledge from relevant domains. This approach can be further enhanced by considering the latent sentiment information. However, as this latent sentiment information is derived from both relevant and irrelevant sources, the performance of the recommendation system may decline. This is a negative transfer (NT) problem, where the knowledge that is derived from multiple sources affects the system. Also, these source domains are often imbalanced, which could further hurt the performance of the recommendation system. To this end, recent research has shown that NT is caused by domain divergence, source and target quality, and algorithms that are not carefully designed to utilise the target data to improve the domain transferability. While various research works have been proposed to prevent NT, these address only some of the factors that may lead to NT. In this paper, we propose a more systematic and comprehensive approach to overcoming NT in sentiment analysis by tackling the main causes of NT. Our approach combines the use of cost weighting learning, uncertainty-guided (aleatoric and epistemic) loss function over the target dataset, and the concept of importance sampling, to derive a robust model. Experimental results on a sentiment analysis task using Amazon review datasets validate the superiority of our proposed method when compared to three other state-of-the-art methods. To disentangle the contributions behind the success of both uncertainties, we conduct an ablation study exploring the effect of each module in our approach. Our findings reveal that we can improve a sentiment analysis task in a transfer learning setting from 4% to 10% when combining both uncertainties. Our outcomes show the importance of considering all factors that may lead to NT. These findings can help to build an effective recommendation system when including the latent sentiment information.

Keywords

Transfer learning, neural networks, bert, uncertainty

1. Introduction

Generally, recommendation systems are used in commercial applications to help users discover the products or services they are looking for. In order to solve the lack of data and the cold-start[1] problem, researchers have increasingly introduced concepts of source domain and target domain into cross-domain recommendation [1]. Through the use of transfer learning, cross-domain based recommendation is able to leverage the rich information from multiple domains as against in a single domain, and transfer knowledge effectively from one domain to another. For cross-domain recommendation to work, however, users' interests or item features must be consistent or correlated across domains [1].

Most existing cross-domain recommendation methods rely only on sharing text information, such as ratings, tags or reviews, and ignore latent sentiment information in the sentiment analysis domain [2]. Recently, methods that consider this latent sentiment information have been proven to be more effective when compared with existing recommendation algorithms that do not consider this information [3]. This is because user reviews are usually subjective, so they would not be able to reflect the user's preferences and sentiments towards different attributes.

As these sentiment data are derived from both relevant and irrelevant sources and the datasets are often imbalanced, the performance of these cross-domain recommendation system may decline due to learning a bias [4]. Also, these cross-domain models did not take into account the bidirectional latent relations between users and items [5]. A better solution to this problem is to introduce transfer learning (TL) [6] into the cross-domain recommendation system [5]. TL systems utilise data and knowledge from a related domain (known as the source domain) to mitigate this learning bias, and can improve the generalizability of models in the target domain [6]. Regrettably, this approach is not always successful un-

DL4SR'22: Workshop on Deep Learning for Search and Recommendation, co-located with the 31st ACM International Conference on Information and Knowledge Management (CIKM), October 17-21, 2022, Atlanta, USA
*Corresponding author.
omondiagbep@landcarereserach.co.nz (O. P. Omondiagbe); sherlock.licorish@otago.ac.nz (S. A. Licorish); stephen.macdonell@aut.ac.nz (S. G. MacDonell)
0000-0002-9267-4832 (O. P. Omondiagbe); 0000-0001-7318-2421 (S. A. Licorish); 0000-0002-2231-6941 (S. G. MacDonell)

[1]A problem where the system cannot draw any inferences for users or items about which it has not yet gathered sufficient information

less specific guidelines are adhered to [7]; 1) both tasks should be related; 2) the source and target domain should be similar; 3) and a model which can learn both domains should be applied to both the source and target datasets. When these guidelines are not followed, the performance of the target model is likely to degrade. This is known as negative transfer (NT) [8]. NT can be caused by four main issues [7]: **One:** Domain divergence - When the divergence between the source and target domains is wide, NT will occur. **Two:** Transfer algorithm - When designing a transfer algorithm, it should have a theoretical guarantee that the performance in the target domain will be better when auxiliary data are used, or the transfer algorithm should be carefully designed to improve the transferability of auxiliary domains, else NT may occur. **Three:** Source data quality - The quality of the source data determines the quality of the transferred knowledge. If the source data are very noisy, then a model trained on them is unreliable. **Four:** Target data quality - The target domain data may be noisy, which may also lead to NT. Also, the amount of labelled target data has a direct impact on the learning process if not fully utilised by the learning algorithm [9, 10].

Various research works have proposed the mitigation of NT, and these are seen in the following areas [7]; **One:** By enhancing the data transferability strategy [11, 7]. This is done by either addressing the domain divergence between the source and target [12, 11], or reweighing strategy by applying more weight to those source domains which are similar to the target dataset [13, 14], or by learning a common latent feature space [15]. **Two:** By enhancing the model transferability enhancement through transferable normalisation [16], or by making the model robust to adversarial samples through the use of a robust optimisation loss function [17]. **Three:** By enhancing the target prediction through the use of pseudo labelling [18, 19].

Previous research found that the use of a model that is robust to adversarial samples results in better transferability [20, 21]. They tend to have better accuracy than a standard target model. Similarly, Liang et al. [20] found a positive correlation between a model that is robust to an adversarial sample and the knowledge transferred. This work suggests such a model can benefit from the knowledge transfer between the source and target. By relying on such methods, these approaches can be limited to being robust to adversarial samples and fail to model uncertainty under data and label distribution, which could introduce further bias [22]. Recently, the work of Grauer and Pei [22] has shown that when model uncertainty is known and distributed evenly, the performance and reliability of the model are greatly improved.

In this work, we introduce the use of an uncertainty-guided loss function to guide the training process when utilising the source and target datasets and incorporate a cost weight to tackle the problem of imbalanced data that may further increase the domain divergence issue. Hence, this work uses the idea of model and data transferability enhancement to develop a more robust model aimed at preventing negative transfer. By using such a systematic approach, we would be able to tackle the four main causes of NT mentioned above. Our main contributions are summarised as follows.

- We propose using a combined uncertainty as a loss function. This combined uncertainty consists of both the aleatoric and epistemic uncertainties. The epistemic uncertainty captures the model uncertainty, while the aleatoric uncertainty captures the uncertainty concerning information that the data cannot explain and is modelled over the target and source dataset to guide the learning process. By using the aleatoric uncertainty-guided loss function over the target and source data, we can derive more information and enhance the model's transferability.
- We propose combining an uncertainty-guided loss function, a cost-sensitive classification method of incorporating cost-weighting into the model and an importance sampling strategy to enhance the data and model transferability. This method can be used when there is imbalanced data and/or dissimilarity between the source and target dataset.
- Finally, we perform an ablation study to disentangle the contributions behind the success of each module introduced in our system.

The remainder of this paper is organised as follows. We present related work in Section 2. Next, we introduce our proposed approach in 3. Section 4 presents our datasets, candidate models, and experimental setup. The results and discussion are presented in Sections 5 and 6, respectively, before considering threats in Section 7. Finally, we conclude the study in Section 8.

2. Related Work

Transfer Learning is a research strategy in machine learning (ML) that aims to use the knowledge gained while solving one problem and apply it to a different but related problem [23]. Early methods in this area have exploited techniques such as instance weighting [24, 25], feature mapping [26, 27] and transferring relational knowledge [28]. Due to the increased processing power afforded by graphical processing units (GPUs), deep learning is now used more frequently in transfer learning tasks and when compared to earlier approaches, such models have

achieved better results in the discovery domain invariant features [29]. It was shown that when deep learning is used the transferability of features decreases as the distance between the base task and target task increases, but that transferring features even from distant tasks can be better than using random features [29]. Some of these deep learning methods [30, 31, 32] have exploited the use of mismatch measurement, such as Maximum Mean Discrepancy (MMD) to transfer features or by using generative adversarial networks (GANs) [33]. Although these methods have all achieved high performance in different domains, such as in computer vision [34] and natural language processing [35], they were not designed to tackle the problem of negative transfer (NT).

Other prominent lines of work can be seen in deep learning to tackle the issue of NT. These works include the use of instance weighting (e.g., predictive distribution matching (PDM) [13]) , enhancing the feature transferability through the use of a latent feature (e.g., DTL [36]), and the use of soft loss function based on soft pseudo-labels (e.g., Mutual Mean-Teaching (MMT)[19]). These methods do not guarantee tackling NT, as they tackle some causes of NT, but not all (e.g., PMD method tackles the transfer algorithm and source data quality, while MMT tackles the domain divergence, transfer algorithm and target data quality issue). Although, a previous study exploring the benefits of modelling epistemic and aleatoric uncertainty in Bayesian deep learning models for vision tasks has demonstrated that when these uncertainties are integrated into the loss functions, the model is more robust to noisy data, how these can be used to tackle NT has not been looked at. **Hence, our main objective in this paper is to derive a robust loss function for deep transfer learning that tackles the causes of NT mentioned in Section 1**.

3. Method

This section provides a formal definition of NT and proposed methods to overcome it.

3.1. Negative Transfer

Notation: We use the following notation $P_s(\mathbf{x}_s) \neq P_t(\mathbf{x}_t)$ and $P_s(y_s|\mathbf{x}_s) \neq P_s(y_t|\mathbf{x}_t)$ to denote the marginal and conditional distribution of source and target sets, respectively. In this case, $\mathbf{x}_s$ and $\mathbf{x}_t$ represent the source and target, respectively. Zhang et al. [7] gave a mathematical definition of NT, and proposed a way to determine the degree of NT (NTD) when it happens.

Definition: Let $\epsilon\varepsilon$ be the test error in the target domain, $\theta(\mathbf{S}, \mathbf{T})$ a TL algorithm between source (**S**) and target (**T**), and $\theta(\emptyset, \mathbf{T})$ the same algorithm which does not use the source domain information at all. Then, NT happens when $\epsilon\varepsilon(\theta(\mathbf{S}, \mathbf{T})) > \epsilon\varepsilon(\theta(\emptyset, \mathbf{T}))$, and the degree of NT can be evaluated by equation 1 below:

$$NTD = \epsilon\varepsilon(\theta(\mathbf{S}, \mathbf{T})) - \epsilon\varepsilon(\theta(\emptyset, \mathbf{T})) \tag{1}$$

When NTD is positive, then negative transfer has occurred. Next, we propose a systematic way to avoid negative transfer.

3.2. Proposed Methods

We explain the three concepts used in our method below:
Cost-sensitive Classification: The idea of cost-sensitive classification is used when there is a higher cost of mislabelling one class over the other class [37]. Cost-sensitive learning tackles the class imbalance problem by changing the model cost function giving more weight to the minority class and multiplying the loss of each training sample by a specific factor. The imbalanced data distribution is not modified directly during training [37]. Madabushi et al. [38] introduced a cost-weighting strategy in the Bert model, which increases the weight of incorrectly labelled sentences by altering the cost function of the final model layer. The cost function is changed by modifying the cross-entropy loss for a single prediction x, and the model's prediction for class k to accommodate an array weight as shown in equation 2

$$loss(x, class) = weight[class]\emptyset \tag{2}$$
$$\text{where } \emptyset = -x[class] + log(\textstyle\sum_{k=1} exp(x[k]))$$

Importance Sampling: The traditional way of training a deep learning model has one major drawback, where it is not able to differentiate samples where it performs very well, i.e., low loss and those samples where the performance is poor i.e., high loss [39]. Also, as not all source samples can provide useful knowledge [39], we introduce the idea of importance sampling to control examples which should be given more priority. Importance sampling [40] is a variance reduction technique and is done by taking a random sample of a set based on a probability distribution among the elements of the group. In our proposed method, we attach weights to the source training examples based on their similarity to the target dataset. The samples with more weight will have a higher chance of being selected. We sample the source from a probability density over the target data.
Uncertainty Quantification: There are different types of uncertainties, and these could be present in the data or model. When the uncertainty is derived from the model, it is referred to as "epistemic or model uncertainty" [41]. Epistemic uncertainty captures the ignorance about the model generated from the collected data and can be explained more when more data is given to the model [41].

It is the property of the model. When the uncertainty is related to the data, it is referred to as aleatoric uncertainty [41]. It captures the uncertainty concerning information that the data cannot explain. This can be further divided into two;

- Heteroscedastic uncertainty, which depends on the data input and is predicted as a model output [41].
- Homoscedastic uncertainty, which is not input data dependent but assumes a constant for all input data and varies between the different tasks [42].

In this case, we are not interested in the homoscedastic uncertainty because we are assuming related task between the source and target. To learn the heteroscedastic uncertainty, the loss function can be replaced with the following [41]:

$$Loss = \frac{||y-\hat{y}||^2}{2\sigma^2} + \frac{1}{2}\log\sigma^2 \quad (3)$$

where the model predicts a mean $\hat{y}$ and variance σ^2.

Kendall and Gal [41] proposed a loss function to combine both epistemic and aleatoric (heteroscedastic) uncertainty as follows:

$$Loss = \frac{1}{D}\sum_{i=1} exp(-\log\sigma^2)||y-\hat{y}||^2 + \frac{1}{2}\log\sigma^2 \quad (4)$$

where D is the total number of output and σ^2 is the variance.

Our Proposed Approach: To derive our proposed loss function, which can enhance the data and model transferability, we combine equation 2 and 3 when incorporating heteroscedastic uncertainty, and equation 2 and 4 when incorporating both epistemic and heteroscedastic uncertainty. To determine the similarity of the sample, we use the method proposed by Kilgarriff [43]. Then, the Wilcoxon signed-rank test [44] is used to compare the frequency counts from both datasets to determine if both datasets have a statistically similar distribution. To overcome the divergence problem, we use the importance sampling technique in our training process. The pseudocode for our proposed method is as follows:

Algorithm 1 Combined Uncertainty Loss Function and Cost-Weighting (CUCW)

Input:

- Source model : $\mathbf{g}(\mathbf{x})$
- Source Training set S_{tr}
- Target Training set T_{tr}
- Target Validation set T_v
- Target Testing set T_{ts}

Output: Degree of negative transfer (NTD)

1. Estimate similarity for each source sample against random 1000 target samples
2. Estimate importance weight with importance sampling based on the similarity
3. Train a source model g using importance weight with a small target sample as the validation data T_v
4. Compute loss function using Equations 2 and 3 OR Equations 2 and 4
5. Compute test error $\epsilon\varepsilon(\theta(\mathbf{S},\mathbf{T}))$ on model g using target test set T_{ts}
6. Train a target model t with the target data only T_{tr}
7. Compute test error $\epsilon\varepsilon(\theta(\emptyset,\mathbf{T}))$ on model t using target test set T_{ts}
8. Calculate NTD $\epsilon\varepsilon(\theta(\mathbf{S},\mathbf{T}))$ - $\epsilon\varepsilon(\theta(\emptyset,\mathbf{T}))$
9. Fine tune model g using target training set T_{tr} and target validation set T_v to derive a new model tg
10. Compute test error $\epsilon\varepsilon(\theta(\emptyset,\mathbf{T}))$ on model tg using target test set T_{ts}
11. **return** Degree of negative transfer (NTD) and model performance

Based on the algorithm above, we can employ a deep transfer learning using the proposed approach to find an optimal model with the least degree of negative transfer. This can be done by following the steps in sequential order. For each step, we can find the best model by training different hyperparameters in our model.

4. Experiments

All experiments were conducted 10 times as used in the work of Bennin et al. [45] to reduce the impact of bias, and the results were averaged across all independent runs. For our sentiment analysis task, we use the Amazon review dataset. We aim to build an accurate sentiment analysis model for low-resource domains by learning from high-resource but related domains. We used the smaller version of the datasets prepared by Lakkaraju et al. [46]. These datasets contain 22 domains, as shown in section 1 above. It is worth noting that some domains in this dataset are imbalanced, as seen in Fig 1. We ranked reviews with 1 to 3 stars as negative, while reviews with 4 or 5 stars were ranked as positive. For the pre-processing steps, we use standard techniques commonly used in NLP and Amazon sentiment analysis tasks [47, 48] in the following order; tokenisation, stop word/punctuation removal, and lemmatisation. Tokenisation involves the process of separating a sentence into a sequence of words known as "tokens" [49]. These tokens are identifiable or

Table 1
Ratio of negative to positive sample in the Amazon datasets

Domains	Ratio
Apparel	0.98
Automotive	1.00
Baby	0.91
Beauty	0.49
Books	0.89
Camera_&_Photo	0.91
Cell_phones_&_Service	0.58
Computer_&_Video_Games	1.00
Dvd	0.96
Electronics	0.91
Grocery	0.34
Health_&_Personal_Care	0.99
Jewelry_&_Watches	0.29
Kitchen_&_Housewares	0.94
Magazines	0.97
Music	1.02
Office_products	0.72
Outdoor_living	0.34
Software	0.63
Sports_&_Outdoors	0.95
Toy_&_Games	0.91
Video	1.24

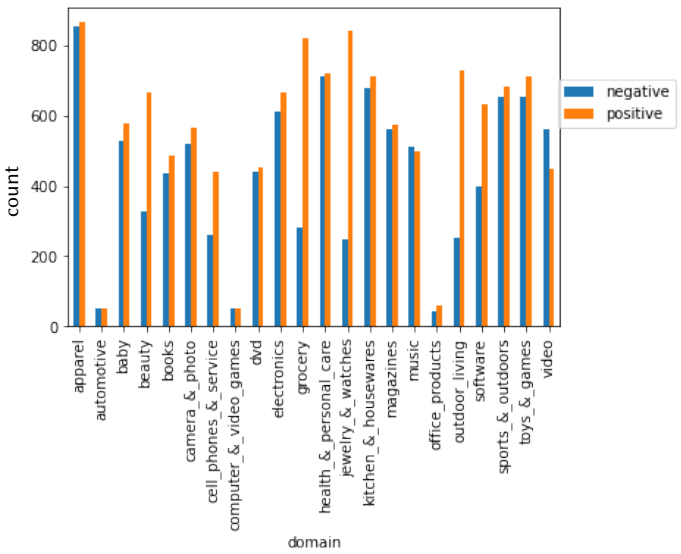

Figure 1: Amazon review dataset

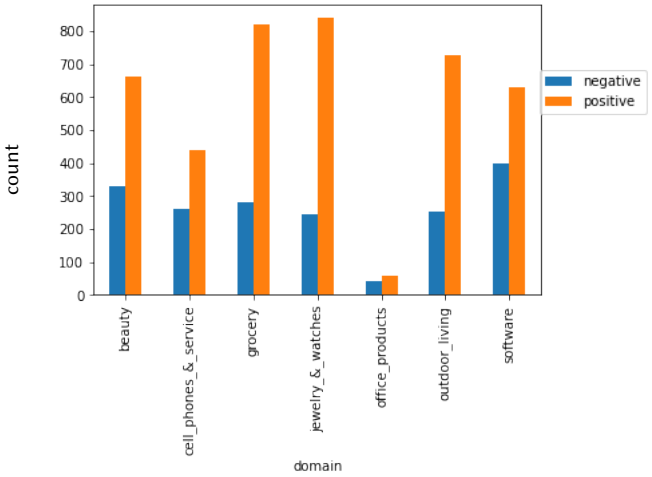

Figure 2: Amazon review dataset showing imbalance domains

separated from each other by a space character. Punctuation and stop words that frequently appear and do not significantly affect meaning (stop word removal e.g., "the", "is" and "and") were also removed [49]. Our lemmatisation process involves using the context in which the word is derived from (e.g., studies becomes study). By lemmatising a word, we reduce its derivationally related forms into a common root form. By using the root form of a word, the model will be able to learn any inflectional form for that given word.

4.1. Experiment Setup

We selected only domains from the Amazon review datasets where class imbalance was evident. To determine the domains to select, the negative to positive ratio is presented in Table 1, where only domains with less than 0.7 ratio were selected to be used in this experiment. From Table 1, six domains were selected as shown in Figure 2 below.

We designed two groups of experiments by selecting domains where class imbalance is present, as shown in Figure 2. In the experiment, we excluded the "Grocery" domain, as this domain is not related to the other six domains shown in Figure 2. The first group of domains consists of datasets from Beauty, Outdoor_living and Jewelry_&_Watches, while the second domain group consists of datasets from Office_products, Cell_phones_&_Service and Software. For each experiment, a single domain was used as the target dataset, while the remaining domains in that group were used as the source datasets.

Text Similarity Measure: We use the Wilcoxon signed-rank test [44] to compare the frequency counts from both datasets to determine if both datasets have a statistically similar distribution. This was done by extracting all words while retaining the repeat from each sample of our

Table 2
BAUC of fine-tuned Bert uncased model and different baseline methods on Amazon review dataset

Groups	Target	CUCW	CUCW No epistemic	CUCW No Importance Sampling	CUCW No Cost Weighting	PDM	MMT	DTL
Group 1	Outdoor living	0.956	0.925	0.942	0.806	0.798	0.810	0.779
✓	Beauty	0.935	0.902	0.921	0.824	0.690	0.745	0.767
✓	Jwellery & Watches	0.931	0.912	0.928	0.776	0.644	0.763	0.745
Group 2	Cellphones & services	0.976	0.956	0.966	0.875	0.789	0.886	0.819
✓	Software	0.965	0.949	0.931	0.854	0.776	0.845	0.788
✓	Office_products	0.957	0.945	0.944	0.818	0.778	0.823	0.799

source training set and ignoring the stop words. From the target set, we sampled (with replacement) 1000 samples as done by Madabushi et al. [38]. Then, we use a word frequency from each of the source training samples and the sample's target set to calculate the p-value using the Wilcoxon signed-rank test.

Model: We used the Bert uncased model for this task. It consists of a 768-dimension vector, 12 layers of the transformer block and 110 million parameters. We added a fully connected layer on top of the BERT self-attention layers to classify the review. For the parameters, we adopt a similar hyperparameter as used in the Bert uncased model for Amazon sentiment analysis [50]. These parameters include using the Adam optimiser with various learning rates and 512 Max Sequence Length with five epochs. The learning rate was 1e−05. The model was first build using the source dataset to derive a source model. Then, this source model was fine-tuned with the target datasets. The fine-tuning with the target datasets was done by using a commonly split ratio (30:70) [51]. The training sets of the target data were used to fine-tune the source model before being tested on the test sets. We ran 10 experiments to compute the estimated risk by the different methods and the average was reported.

Evaluation measures: All experiments were conducted 10 times as done in the work of Bennin et al. [45] to reduce the impact of sampling bias, and the results were averaged across the independent runs. To evaluate the prediction accuracy of each modelling approach, the following were computed:

- **Balanced accuracy (BAUC):** BAUC measures model performance, taking into account class imbalances and it also overcomes bias in binary cases [52]. The balanced accuracy is computed as the average of the proportion of correct predictions for each class separately.
- **F-measure:** This is used for evaluating binary classification models based on the predictions made for the positive class [52].

5. Results

Here, we compare our systematic approach against three different strategies proposed for tackling NT. These strategies were:

- Predictive distribution matching (PDM) [13]. This is an instance-based weighting approach. This method works by first measuring the differing predictive distributions of the target domain and the related source domains. In this case, a PDM regularised classifier is used to infer the target pseudolabeled data, which will help to identify the relevant source data, so as to correctly align their predictive distributions [13]. We used the support vector machines (SVM) variant of the proposed PDM as used in the sentiment analysis task in the work of Seah et al. [13].
- Mutual Mean-Teaching (MMT)[19]: This is a feature transferability approach which uses a soft loss function based on soft pseudo-labels and is carried out in two stages. In the first stage, the Bert uncased model was trained using the source domain to derive a source model. This source model is trained to model a feature transformation function that transforms each input sample into a feature representation. For this experiment. the source model is trained with a classification loss and a triplet loss to separate features belonging to different identity, as used in the original paper [19]. Next, the source model trained in stage 1 is optimised using the MMT framework, which is based on the clustering method. The details of this approach are explained in the original paper [19].
- Dual Transfer Learning (DTL) [15]: This approach enhances feature transferability through the use of a latent feature. This method simultaneously learns the marginal and conditional distributions, and exploits their duality. For this experiment, the training was done using the Bert

Table 3
F-measure of fine-tuned Bert uncased model and different baseline methods on Amazon review dataset

Groups	Target	CUCW	CUCW No epistemic	CUCW No Importance Sampling	CUCW No Cost Weighting	PDM	MMT	DTL
Group 1	Outdoor living	0.945	0.911	0.903	0.800	0.778	0.808	0.756
✓	Beauty	0.922	0.899	0.909	0.799	0.665	0.716	0.733
✓	Jwellery & Watches	0.898	0.886	0.886	0.730	0.616	0.742	0.709
Group 2	Cellphones & services	0.965	0.931	0.961	0.832	0.742	0.835	0.799
✓	Software	0.949	0.919	0.925	0.818	0.754	0.778	0.718
✓	Office_products	0.949	0.927	0.939	0.832	0.731	0.809	0.817

Table 4
BAUC of none fine-tuned Bert uncased method and different baseline methods on Amazon review dataset

Groups	Target	CUCW	CUCW No epistemic	CUCW No Importance Sampling	CUCW No Cost Weighting	PDM	MMT	DTL
Group 1	Outdoor living	0.887	0.845	0.864	0.799	0.798	0.810	0.779
✓	Beauty	0.935	0.902	0.921	0.824	0.690	0.745	0.767
✓	Jwellery & Watches	0.853	0.821	0.843	0.734	0.644	0.763	0.745
Group 2	Cellphones & services	0.939	0.909	0.920	0.840	0.789	0.886	0.819
✓	Software	0.915	0.898	0.878	0.819	0.776	0.845	0.788
✓	Office_products	0.919	0.888	0.865	0.843	0.778	0.823	0.799

uncased model by combining the source and target training data before being tested on the target dataset.

In Tables 2 to 3, we report the fine-tuned models' performance (balanced accuracy and F- measure) on the target test set. In cases where NT has occurred (i.e., the degree of NT was calculated using Equation 1), we denote the colour of the accuracy as red. From Table 2, the results indicate that our proposed approach with fine-tuning, other components and including both uncertainties (heteroscedastic aleatoric and epistemic uncertainty) in the loss function outperformed the other three models. To disentangle the contribution of all components in our proposed approach, we report the results by removing each component in our proposed approach. When epistemic uncertainty or cost weighting was excluded from the loss function, we noticed three cases (i.e., outdoor living, cell phones & service, and office product were used as the target datasets) where the MMT method outperformed our approach. A similar outcome was noted in the F-measure as shown in Table 3. To further disentangle the contribution of all components in our proposed approach without fine-tuning the Bert model and to provide a fair comparison with the three methods we compared against, we combined the source and target training data to train our Bert model before testing on the target test data. This

Table 5
F-measure of the none fine-tuned Bert uncased method and different baseline methods on Amazon review dataset

Groups	Target	CUCW	CUCW No epistemic	CUCW No Importance Sampling	CUCW No Cost Weighting	PDM	MMT	DTL
Group 1	Outdoor living	0.881	0.844	0.829	0.770	0.778	0.808	0.756
✓	Beauty	0.899	0.865	0.834	0.788	0.665	0.716	0.733
✓	Jwellery & Watches	0.822	0.808	0.789	0.710	0.616	0.742	0.709
Group 2	Cellphones & services	0.887	0.858	0.887	0.787	0.742	0.835	0.799
✓	Software	0.878	0.844	0.868	0.819	0.754	0.778	0.718
✓	Office_products	0.843	0.822	0.829	0.709	0.731	0.809	0.817

was done to remove the benefit of the fine-tuning component in our design. The results in Tables 4 to 5 show that, without the fine-tuning component, we were still able to improve the performance when all other components are integrated in our deep transfer learning, but with less improvement (i.e., noting an improvment of BAUC and F-measure of 2% to 9% as shown in Table 4 and Table 5).

6. Discussion

In our sentiment analysis experiment (see Table 2 to Table 5), our proposed method, which incorporated both uncertainties, was able to improve the balanced accuracy of the BERT model from 5% to 14% and F- measure value from 5% to 10% as compared to using techniques that are instance [13] or feature transferability based [19, 15]. Although the instance level transferability enhancement has been used in the deep learning model to prevent NT [11, 53], they do not handle the target data quality. This factor is shown to be one of the causes of NT [7]. The PDM method that we compared against in this paper tackles the domain divergence issue by using predictive distribution matching to remove the irrelevant source. This method still failed to address the target data quality; hence, we noted a single case of nt in our nlp task result (when the outdoor living domain was used as the target's dataset). Although the MMT method uses a softmax loss function based on soft pseudo-labels to tackle the target data quality, it cannot tackle the domain divergence issue, which may also lead to NT. A single case of NT (when the outdoor living domain was used as the target's dataset) was also noted when using this method. On the other hand, our proposed method is more robust. It uses the uncertainty-guided function to tackle the target and source data quality issue, importance sampling and cost weighting learning, to tackle the domain divergence problem. For the fine-tuning process, we use a small target sample as the validation data in the source model to improve the transferability of the final model. Our results show that the final model is improved when we introduce the use of an uncertainty-guided loss function to guide the training process when utilising the source and target datasets and incorporate a cost weight to tackle the problem of imbalanced data. In the work of Grauer and Pei [22], it was also noted that when model uncertainty is known and distributed evenly, the performance and reliability of the model are greatly improved. Hence, this work uses the idea of model and data transferability enhancement to develop a more robust approach aimed at preventing negative transfer. The evidence from our results suggests that we could use a systematic approach such as what was proposed in this paper to improve the quality of models in a deep transfer learning setting. Also, it is worth noting that two of the methods we compared against (MMT and DTL) in this study also use the Bert Uncased model, hence, we are able to eliminate the interference of model complexity in the comparison result. From the ablation study, model fine-tuning improved the overall performance from 2% to 6% when integrating all components into our approach.

7. Addressing threats to validity

The experimental dataset was compiled by [46]. We acknowledge threats relating to errors in the review labels. These threats have been well minimised by experimenting with different projects in the datasets. Also, we concede that there are a few uncontrolled factors that may have impacted the experimental results in this study. For instance, there could have been unexpected faults in the implementation of the approaches we compare against in this paper [54]. We sought to reduce such threats by using the source code provided for these methods (e.g., PDM, MMT and DTL). While we recognize the threats above, we anticipate that our study here still contributes novel findings to transfer-based modelling for recommendation systems in NLP domains relying on latent sentiment information.

8. Conclusion

In this work, we proposed a systematic approach to overcoming negative transfer by tackling domain divergence, taking account of the source and target data quality. Our approach involves using cost weighting learning, uncertainty-guided loss function over the target dataset, and the concept of importance sampling to derive a robust model. This systematic approach improves the target domain's performance. The results reported in this work also reveal that when both aleatoric heteroscedastic and epistemic uncertainty are combined, we can further enhance the performance of the target model. We therefore assert that our systematic approach is a good approach for overcoming negative transfer and improving target model performance when performing sentiment analysis in a transfer learning setting. This approach can be used to build an effective recommendation system when including the latent sentiment information. A plausible next step, is to use such an approach to design an effective recommendation system that takes into account the latent sentiment information. Although our experiments showed our approach improves the target model performance and prevents NT in sentiment analysis, it is still important to investigate this approach for other domains.

Acknowledgements

This research was partly supported by an Internal Research fund from Manaaki Whenua — Landcare Research, New Zealand. Special thanks are given to the Department of Informatics at Landcare Research for their ongoing support.

References

[1] P. Cremonesi, A. Tripodi, R. Turrin, Cross-domain recommender systems, in: 2011 IEEE 11th International Conference on Data Mining Workshops, Ieee, 2011, pp. 496–503.

[2] T. Zang, Y. Zhu, H. Liu, R. Zhang, J. Yu, A survey on cross-domain recommendation: taxonomies, methods, and future directions, arXiv preprint arXiv:2108.03357 (2021).

[3] Y. Wang, H. Yu, G. Wang, Y. Xie, Cross-domain recommendation based on sentiment analysis and latent feature mapping, Entropy 22 (2020) 473.

[4] B. Zadrozny, Learning and evaluating classifiers under sample selection bias, in: Proceedings of the twenty-first international conference on Machine learning, 2004, p. 114.

[5] P. Li, A. Tuzhilin, Ddtcdr: Deep dual transfer cross domain recommendation, in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 331–339.

[6] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (2009) 1345–1359.

[7] W. Zhang, L. Deng, L. Zhang, D. Wu, A survey on negative transfer, 2020. URL: https://arxiv.org/abs/2009.00909. doi:10.48550/ARXIV.2009.00909.

[8] M. Rosenstein, Z. Marx, L. Kaelbling, & dietterich, tg (2005). to transfer or not to transfer, in: NIPS 2005 Workshop on Transfer Learning, ????

[9] Z. Wang, Z. Dai, B. Póczos, J. Carbonell, Characterizing and avoiding negative transfer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11293–11302.

[10] O. P. Omondiagbe, S. Licorish, S. G. MacDonell, Improving transfer learning for cross project defect prediction, TechRxiv preprint techrxiv.19517029 (2022).

[11] Z. Wang, J. Carbonell, Towards more reliable transfer learning, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2018, pp. 794–810.

[12] E. Eaton, et al., Selective transfer between learning tasks using task-based boosting, in: Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.

[13] C.-W. Seah, Y.-S. Ong, I. W. Tsang, Combating negative transfer from predictive distribution differences, IEEE transactions on cybernetics 43 (2012) 1153–1165.

[14] D. Wu, Pool-based sequential active learning for regression, IEEE transactions on neural networks and learning systems 30 (2018) 1348–1359.

[15] M. Long, J. Wang, G. Ding, W. Cheng, X. Zhang, W. Wang, Dual transfer learning, in: Proceedings of the 2012 SIAM International Conference on Data Mining, SIAM, 2012, pp. 540–551.

[16] X. Wang, Y. Jin, M. Long, J. Wang, M. I. Jordan, Transferable normalization: Towards improving transferability of deep neural networks, Advances in neural information processing systems 32 (2019).

[17] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, arXiv preprint arXiv:1706.06083 (2017).

[18] L. Gui, R. Xu, Q. Lu, J. Du, Y. Zhou, Negative transfer detection in transductive transfer learning, International Journal of Machine Learning and Cybernetics 9 (2018) 185–197.

[19] Y. Ge, D. Chen, H. Li, Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification, arXiv preprint arXiv:2001.01526 (2020).

[20] K. Liang, J. Y. Zhang, O. O. Koyejo, B. Li, Does adversarial transferability indicate knowledge transferability? (2020).

[21] Z. Deng, L. Zhang, K. Vodrahalli, K. Kawaguchi, J. Y. Zou, Adversarial training helps transfer learning via better representations, Advances in Neural Information Processing Systems 34 (2021).

[22] J. A. Grauer, J. Pei, Minimum-variance control allocation considering parametric model uncertainty, in: AIAA SCITECH 2022 Forum, 2022, p. 0749.

[23] R. Caruana, D. Silver, J. Baxter, T. Mitchell, L. Pratt, S. Thrun, Learning to learn: knowledge consolidation and transfer in inductive systems, in: Workshop held at NIPS-95, Vail, CO, see http://www. cs. cmu. edu/afs/user/caruana/pub/transfer. html, 1995.

[24] M. Sugiyama, M. Krauledat, K.-R. Müller, Covariate shift adaptation by importance weighted cross validation., Journal of Machine Learning Research 8 (2007).

[25] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, A. Smola, Correcting sample selection bias by unlabeled data, Advances in neural information processing systems 19 (2006).

[26] T. Jebara, Multi-task feature and kernel selection for svms, in: Proceedings of the twenty-first international conference on Machine learning, 2004,

p. 55.
[27] S. Uguroglu, J. Carbonell, Feature selection for transfer learning, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 430–442.
[28] L. Mihalkova, R. J. Mooney, Transfer learning by mapping with minimal target data, in: Proceedings of the AAAI-08 workshop on transfer learning for complex tasks, 2008.
[29] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, Advances in neural information processing systems 27 (2014).
[30] B. Sun, K. Saenko, Deep coral: Correlation alignment for deep domain adaptation, in: European conference on computer vision, Springer, 2016, pp. 443–450.
[31] M. Long, Y. Cao, J. Wang, M. Jordan, Learning transferable features with deep adaptation networks, in: International conference on machine learning, PMLR, 2015, pp. 97–105.
[32] G. K. Dziugaite, D. M. Roy, Z. Ghahramani, Training generative neural networks via maximum mean discrepancy optimization, arXiv preprint arXiv:1505.03906 (2015).
[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27 (2014).
[34] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, R. Chellappa, Generate to adapt: Aligning domains using generative adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8503–8512.
[35] W. Y. Wang, S. Singh, J. Li, Deep adversarial learning for nlp, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials, 2019, pp. 1–5.
[36] M. Rajesh, J. Gnanasekar, Annoyed realm outlook taxonomy using twin transfer learning, International Journal of Pure and Applied Mathematics 116 (2017) 549–558.
[37] M. Kukar, I. Kononenko, et al., Cost-sensitive learning with neural networks., in: ECAI, volume 15, Citeseer, 1998, pp. 88–94.
[38] H. T. Madabushi, E. Kochkina, M. Castelle, Cost-sensitive bert for generalisable sentence classification with imbalanced data, arXiv preprint arXiv:2003.11563 (2020).
[39] A. Katharopoulos, F. Fleuret, Not all samples are created equal: Deep learning with importance sampling, in: International conference on machine learning, PMLR, 2018, pp. 2525–2534.
[40] C. P. Robert, G. Casella, G. Casella, Monte Carlo statistical methods, volume 2, Springer, 1999.
[41] A. Kendall, Y. Gal, What uncertainties do we need in bayesian deep learning for computer vision?, Advances in neural information processing systems 30 (2017).
[42] Q. V. Le, A. J. Smola, S. Canu, Heteroscedastic gaussian process regression, in: Proceedings of the 22nd international conference on Machine learning, 2005, pp. 489–496.
[43] A. Kilgarriff, Comparing corpora, International journal of corpus linguistics 6 (2001) 97–133.
[44] F. Wilcoxon, Probability tables for individual comparisons by ranking methods, Biometrics 3 (1947) 119–122.
[45] K. E. Bennin, J. Keung, P. Phannachitta, A. Monden, S. Mensah, Mahakil: Diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction, IEEE Transactions on Software Engineering 44 (2017) 534–550.
[46] H. Lakkaraju, J. McAuley, J. Leskovec, What's in a name? understanding the interplay between titles, content, and communities in social media, in: Proceedings of the International AAAI Conference on Web and Social Media, volume 7, 2013, pp. 311–320.
[47] A. S. AlQahtani, Product sentiment analysis for amazon reviews, International Journal of Computer Science & Information Technology (IJCSIT) Vol 13 (2021).
[48] A. F. Anees, A. Shaikh, A. Shaikh, S. Shaikh, Survey paper on sentiment analysis: Techniques and challenges, EasyChair2516-2314 (2020).
[49] D. D. Palmer, Tokenisation and sentence segmentation, Handbook of natural language processing (2000) 11–35.
[50] M. Geetha, D. K. Renuka, Improving the performance of aspect based sentiment analysis using fine-tuned bert base uncased model, International Journal of Intelligent Networks 2 (2021) 64–69.
[51] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann Publishers, San Francisco, 2016. doi:10.1016/c2009-0-19715-5.
[52] T. Menzies, J. Greenwald, A. Frank, Data mining static code attributes to learn defect predictors, IEEE transactions on software engineering 33 (2006) 2–13.
[53] Y. Xu, H. Yu, Y. Yan, Y. Liu, et al., Multi-component transfer metric learning for handling unrelated source domain samples, Knowledge-Based Systems 203 (2020) 106132.
[54] E. A. Felix, S. P. Lee, Predicting the number of defects in a new software version, PloS one 15 (2020) e0229131.