A Transferable, Cross-Domain Sentiment Analysis Model

1. Introduction

Sentiment analysis is a branch of Natural Language Processing that deals with interpreting the attitude of a text towards a certain topic. Sentiment analysis models aim to predict the polarity of a given text, either on a discrete (positive vs. negative) or continuous (rating 0-10) scale. While there exists several large labelled datasets (Amazon product reviews, IMDB movie reviews, YELP reviews, etc.), much of the demand for sentiment analysis comes in domains outside of those with large labelled datasets. As a result, a model that is capable of taking an existing labelled dataset to create a general model capable of adapting to a domain with a smaller set of labelled data would be extremely useful for these scenarios. This project aims to implement an analyze a model created for this purpose.

2. Related Work and Critical Review

2.1. Domain-Aware BERT Summary

Natural Language Processing is limited to some degree due to the fact that there aren't massive amounts of labeled data for every task and domain. This paper seeks to address this problem by increasing the transferability of NLP models by training on a source domain that does have adequate labeled data, and then transferring that model to a target domain that lacks labeled data (Du et al., 2020). The particular focus on this paper within NLP is sentiment analysis - predicting sentiment polarity on a target domain using a model trained on a source domain. Deep neural networks for sentiment analysis require a vast amount of labelled training data, which is only available for specific applications, but this paper explores the possibility of reusing the labelled data in one domain and applying the knowledge obtained from the original domain to a new domain that lacks this labelled data.

This model implements a modified version of Bidirectional Encoder Representations from Transformers (BERT). BERT lacks domain awareness, which is essential for this implementation, so modifications were made to the original implementation of BERT. The first addition is a pre-training application on adapting to the target domain's unsupervised data. A second addition created a post-training procedure to promote domain awareness. The pre-training addition involved feature-based learning - contextualizing work representations - and fine-tuning - which prepared the model for adapting to the unsupervised domain objective. The post-training model seeks to inject target domain knowledge into BERT, to optimally differentiate between the source and target domains. The natural language inference aspect of BERT, which includes next sentence prediction capabilities, was removed due to its lack of relevance to this model, but added domain distinguishing capabilities in its place (Du et al., 2020).

Adversarial training was applied to features determined to be domain-independent - appearing in both the source and target domains. The adversarial training included implementing a fully-connected sentiment classifier, and a domain discriminator, which predicted domain labels of samples using a deep classification model.

The model was fed source domain information, which included labeled and unlabeled samples from the source domain - sentences with or without their polarity, as well as unlabeled target domain sentences. The model was trained using the Amazon Dataset, specifically the domains of books, DVDs, electronics, and kitchen appliances to establish separate domains to determine how effectively the model could transfer its training from one domain to another. In order to achieve domain recognition, the model was also fed input sentence pairs - a sentence from the source domain with either another source domain sentence or a target domain sentence, and then labelled based on whether the sentences were from the same domain, training the implementation of BERT to gain domain awareness (Du et al., 2020).

This implementation was compared to 5 other state-of-the-art methods, and was also tested using several variants of BERT. The variant of BERT described in the paper outperformed all other variants, and the model as a whole outperformed the other state-of-the-art methods, providing steps toward a solution for the transferability of sentiment learning from domains with labelled training data to domains without such resources.

2.2. Domain-Aware BERT critical review

The Cross-Domain Sentiment Analysis concept has the potential to be extremely useful in the domain of Natural Language Processing, and this paper clearly explains the relevance of this concept. Additionally, the paper includes very clear and relevant visual representations of the results - including multiple graphs of different types to elaborate on where the implementation is strong and where it may be lacking (Du et al., 2020). The description of the implemen-

tation was also very thorough, with a detailed description of their model and the changes that were made to BERT to optimize it for this particular task. However, the paper makes it hard to distinguish between the pre-training and post-training procedures, and it was difficult to keep track of what the purpose of each was. Additionally, the paper doesn't elaborate much on how similar or different the results were between their implementation of BERT and other implementations. Aside from providing raw metric outcomes, it is hard to tell how significant the improvement was with this particular implementation compared to the other BERT varitations it was tested against.

2.3. ELMo and DistilBERT Comparison Summary

Over the past few years, several state-of-the-art deep contextual language representation models have emerged, two of the highest performers being ELMo and DistilBERT. This paper seeks to explore the potential of building generalizable Natural Language Processing Systems by comparing the two methods on their ability to perform supervised learning on classification and sentiment analysis tasks, as well as transfer their supervised models on source domains to target domains (Büyüköz et al., 2020).

Two datasets are utilized for the implementation and evaluation of these methods - the first being a set of local news articles from China and India (in English) used for binary classification of whether the articles are focused on protests or not, and the second being sentiment analysis data including movie reviews and customer reviews. The methods are evaluated both on their ability to analyze these domains but also to transfer knowledge learned by the source domain to a target domain. The methods were trained on news articles from India and transferred to classify news articles from China, as well as trained on movie reviews and transferred to identify the polarity of customer reviews.

Extensive evaluation was conducted on the methods to determine reliable, wide-reaching results. Testing was done on the source and target domains for both classification and sentiment analysis for both ELMo and DistilBERT, producing 8 total metrics for each implementation variant (Büyüköz et al., 2020). Four variants were tested - frozen embeddings, Feed-Forward Neural Network, external contextualization via BiLTSM, and a combined FFNN and BiLTSM. These tests pointed to several important conclusions.

DistilBERT is significantly smaller and nearly twice as fast compared to ELMo, and this showed during the testing, as there were several instances where ELMo could only handle 150 tokens per input, compared to the standard 256, impacting its peak performance ceiling. In general, the two methods were comparable on the source domain testing, but DistilBERT largely outperformed ELMo on its ability to transfer over to the target domains tested. Overall, DistilBERT edges out ELMo in its cross-domain analysis capabilities, and the added benefits of smaller size and faster training further increase its appeal over ELMo.

2.4. ELMo and DistilBERT Comparison Critical Review

The comparison of these two methods involved extremely thorough experimentation - implementing transferability testing in two different NLP tasks (classification and sentiment analysis) and used four different testing implementations for each (Büyüköz et al., 2020). The results were summarized in full detail for both ELMo and DistilBERT for both tasks on both the source domain and target domain, and the size and training time for both methods were well documented. Additionally, two other comparable methods were implemented and tested for reference. Like the previous paper, this paper did not make it clear how significant the performance differences were between ELMo and DistilBERT. It was very clear that DistilBERT was smaller and faster, but it wasn't nearly as clear how significant the performance differences were. They seemed to perform quite similarly on the source datasets, and DistilBERT clearly outperformed on the transfer tasks to some degree, but it was hard to get a sense of how much of an improvement in performance it really had on ELMo. It also did not very clearly explain the political dataset. The paper mentions this data several times but did not make it clear the structure or purpose of this data, and only after several read-throughs of the description did the gist of the dataset start to make sense. The data is very relevant to the paper, it just needs to be explained more clearly the first time. Overall, the paper clearly describes a comparison and evaluation of two state-of-the-art deep contextual language representations models.

2.5. Cross-Domain Sentiment Classification Summary

Natural Language Processing faces an issue that reviews span so many domains that it is hard to get annotated training data for every type, so it is hard developing models specifically for categories with as little labeled data as possible. This paper seeks to address this issue by proposing a model that can be trained on a source domain with labeled training data and then transferred to a target domain with minimal labeled training data. Most current solutions to this issue use domain-invariant features and disregard information that is specific to a particular domain of interest. This model (Peng et al., 2018) proposes a method that creates two classifiers to account for both domain-invariant and domain-specific features.

In order to achieve this proposed model, the domain-independent classifier is trained using data from both the source and target domains, but the domain-specific classifier

is trained specifically on the target domain data. The classifiers are central moment discrepancy (CMD) based, which involves measuring the difference between probability distributions of two high-dimensional random variables, to create a regularizer to extract representations of the target domain using both domain-specific and domain-independent classification (Peng et al., 2018). Compensation for a lack of labeled target domain data is addressed using a co-training-based system, where confidently predicted unlabeled data from the target domain is transferred to the training set, boosting the amount of labeled data without requiring labeling to be done manually.

Experimentation of this method was done on the Amazon Dataset, which includes 4 domains of books, DVDs, electronics, and kitchen appliances, totalling 12 cross-domain classification tasks with evenly split source domain labeled/unlabeled data and 50 labeled target domain data to simulate minimal availability of labeled data in the target domain. Implementation of this method on these tasks yielded results that outperformed previous state-of-the-art models.

For the purposes of sentiment analysis, reviews are classified as positive or negative based on how many stars were received (3 or lower being negative and above 3 being positive). A TF-IDF matrix was created from all of the reviews. Because there are four different product classifications within the dataset, 12 possible cross-combination pairings are possible, with each having 2000 labelled samples from the source domain, 2000 unlabeled samples from the target domain as well as 50 labelled samples from the target domain. The model then uses the labelled samples and the created model to apply to the labelled target samples to adapt the model to the unlabeled target samples.

2.6. Cross-Domain Sentiment Classification Critical Review

This paper (Peng et al., 2018) describes a unique approach to solving a relevant Natural Language Processing problem and clearly identifies elements of the model that allow it to excel - particularly the use of 2 classifiers, including one specific for domain-specific classification. Previous methods only focused on domain-independent analysis, which leaves out a significant amount of valuable insight, and this model identified a way to include both. The paper also includes a good visual representation of the processes that the method describes, particularly Figure 2 and Algorithm 1, both of which help the reader understand the high level logic behind the model and get a sense of how this can be implemented in code. One thing that can be improved upon is the paper fails to give a clear indication of how much or little labeled data in the target domain is actually needed to produce successful results. It states that 50 samples are used in their experimentation, but does not describe why this number was chosen or whether more data would improve classification results or less data could be used to achieve similar results.

Another critique of the paper is in the form of how the dataset is used. The Amazon reviews are classified positive or negative based on how many stars are given - 3 or lower being classified as negative and higher than 3 classified as positive. This means that every review is classified as either positive or negative, even reviews that aren't particularly polar, which can make the ability to classify certain cases more difficult when there is not much of a difference between positive and negative polarity in some cases. This dataset is different than the IMDB dataset, for example, which is highly polarized, only selecting reviews 7/10 or higher and 4/10 or lower for the positive and negative polarity, removing potential discrepancies associated with neutral reviews. Using the Amazon dataset in this case may make training and model creation a bit more difficult because of this.

3. Implementation Details

3.1. Existing Code Summary

The Cross-Domain Sentiment Classification Paper (Peng et al., 2018) has a repository associated with the methods described in the paper. The first step of implementation involved analyzing and implementing the existing source code from this repository. The repository included a few code scripts as well as the Amazon dataset in the form of a MAT file. One of the Python files is for creating the base model, or CoTraining object, in a very computationally heavy process with a lot of labelled data. At the end of this process a CoTrain model is created and associated items are stored in pickle files for future use.

This model can then be accessed by the other classes in the repository after the initial model is created. This is done during the transfer process, where the original model is used, and then adapted from the source domain to the target domain using a small amount of labelled data from the target domain. This process allows small datasets from domains without a lot of labelled data to still be analyzed in large quantities for sentiment analysis purposes.

3.2. Extension of Existing Repository

Implementation for this project was primarily based around the repository associated with the model discussed previously (Peng et al., 2018). This model focuses on addressing the issue of not having vast amounts of labelled data for every NLP problem, specifically in the field of sentiment analysis, by using a model trained on a large data, adapting it to a different domain using a significantly smaller subset of labelled data from the new domain to train a new model

for this other sentiment analysis application. The repository associated with this paper is created using Python, and TensorFlow v1, which is a couple years outdated as TensorFlow 2 was released a couple years ago. The tentative initial plan was to implement this repository exactly as it was, run the various aspects of the model, and then reimplement it in TensorFlow 2 and PyTorch, as well as test it on datasets beyond the Amazon dataset that was originally used.

The model is extremely complex, with several different components, and the initial model creation is done in a multi-stage training process, and given an inexperience with both Natural Language Processing techniques and TensorFlow, many things were difficult to understand at first. Everything could be reimplemented in code that was compatible with TensorFlow2, primarily by simplifying expressions and replacing tf with tf.compat.v1 when necessary (for example, session and graph initialization, since these were removed in TF2). At this point the code was runnable, and everything could be tested, creating the base model by using the segment of the Amazon dataset of labelled book reviews to determine the outcome of DVD review polarity. This model took 8-12 hours to train on CPU and over 4 hours when implemented in Google Colab with the default GPU. This was expected as is the case with many deep learning problems with large complex code and massive datasets.

In order to determine the effectiveness of the model in the scope of the paper, the model needed to be applied to other sentiment analysis domains using just small sections of their labelled data, which could be done using the provided Amazon dataset, which included electronics and kitchen product reviews along with the book and DVD reviews used to create the original model. A separate python script was used to take the original model and apply it to 3 other combinations of transfer learning (using book reviews to predict kitchen product reviews, using electronics reviews to predict kitchen product reviews, and using electronics reviews to predict DVD review outcomes). The combined creation of each of these three models takes less than 30 minutes to run using CPU, which shows the power of the transfer learning aspect of the model, and verifying the ability to use a large labelled dataset to create a model that can be applied to data without as much labelled cases. Intentions were to extend the model to test on the IMDB dataset as well, however differences in the formatting of these two datasets created large challenges with reading in and using the IMDB data in the same way as the Amazon data so that it could be used for the purposes of the model, so instead just combinations of the four different classifications of the Amazon products were used for testing (books, DVDs, electronics, kitchen).

The original plan was to reimplement the model in PyTorch, using the TensorFlow code and paper as a guideline and figuring it wouldn't be too difficult to translate the TensorFlow code into PyTorch. However, the repository was written in TensorFlow v1 code which had several components that don't exist in TensorFlow 2 and PyTorch. The first goal was to rewrite the TF v1 code in v2, getting rid of the sessions and Graph implementations that were implicit in v2. This proved to be a very tedious process, as there is not much documentation for actually translating v1 code into v2 by getting rid of sessions and graphs (most just recommend using tf.compat.v1, which worked but wouldn't really be useful in helping with the original goal of rewriting the code in PyTorch, which also doesn't have sessions and graph initialization calls). After removing elements of the code that don't exist in TensorFlow 2, there were several errors that occurred when running the model, primarily associated with monitoring the gradients of the respective variables. These issues were likely due to a combination of inexperience with TensorFlow and NLP, as well as the nature of TF v1 and sessions themselves and the complexities associated with using them. After quite a bit of struggling to figure out these errors, it was decided that it would make most sense to leave the sessions in the code and focus on expanding on the model and analyzing its output and effectiveness in the implementation.

The original model creation code and dataset handling code were left untouched, as the primary goal for this implementation was focused on the transferring and plotting aspects of the paper. The dataset handler read in the Amazon dataset, which was in the form of a .mat file with approximately 30000 labelled entries. A TF IDF was created as the model was implemented, and each corpus of data was split into several sections - 8 total with source/target, x/y, and train/test differentiations. The test data was then further split into tuning and validation sections, in addition to the test sections. The transfer script used a very small segment (50 samples) of labelled data to retrain a model based on the original, addressing a scenario where labelled data is not present.

3.3. Experimentation with Model

Once the base model finished training, the next step was to optimize various parameters associated with the transfer model so that the source domain could map to the target domain as efficiently as possible, effective in its ability to predict polarity but also simple enough to train on a minimal amount of labelled data in the target domain and ideally minimized training time as well.

Tweaking of model parameters was done in several phases. The first involved how many corpuses of data would be used for optimal training. The full dataset contains 30,000 samples, which are divided into segments of 5,000 each. The following table illustrates the effectiveness of complete models used with the full 30,000 samples compared to just

one corpus of 5,000 samples for each of the four scenarios described previously (note: all other parameters from the original model were unchanged).

5,000 samples (1 corpus)	30,000 samples (6 corpuses)
0.8102767	0.8270751
0.829101	0.83836883
0.77140975	0.76910406
0.87099165	0.8746988

Table 1: Different number of corpuses for training

Just one of the four instances saw an increase in more than 1 percent effectiveness, and one even saw a decrease. Therefore, the number of 5,000-sample subsets of the data used to train the model don't seem to be extremely important, as long as there is enough data to satisfy what is needed for the 5,000-sample corpus.

The next parameter experimentation and tweaking was focused on breaking down the optimal size of the data corpus used for training. The original model used a size of 5k samples for training, but testing was done to determine whether a smaller size could achieve similar results. Scaling down the corpus size also involved scaling down every associated component used in model generation, for example if the corpus size was cut down 50 percent, then the test, train, and validation groups for x and y in both the source and target domains were all reduced by 50 percent so that the dimensions of everything matched up and that the entire corpus was used. In total, four corpus sizes were tested (5k, 2k, 1k, and 500), all with their corresponding parameter values modified accordingly.

5k	2k	1k	500
0.8102767	0.80421865	0.47676498	0.4476247
0.829101	0.8050655	0.47944403	0.46820027
0.77140975	0.70558524	0.5113941	0.45000994
0.87099165	0.8494323	0.45852232	0.46955344

Table 2: Different corpus sizes

As described in the above table, the results of this experimentation were interesting. 2,000-sample corpus training and testing yielded results not as effective as the 5k size, but there wasn't much of a drop off (0-7 percent drop for each of the four scenarios). However, the 1,000-sample corpus results yielded significant drop off from both the 5k and 2k sizes. The effectiveness fell from 77-87 percent accuracy at 5k to 70-85 percent accuracy at 2k all the way to 45-51 percent with 1,000 samples. 500-sample corpus yielded even worse results, although not a huge difference compared to the 1k, all falling between 44 and 47 percent accuracy. Overall it would appear that sticking with a larger corpus, such as the 5k-sample corpus from the original model, will yield the most effective results. However if there is not as much data available the corpus can be reduced over 50 percent to still yield slightly reduced accuracy, but dropping below 2,000 samples would negatively effect the accuracy of the model significantly

Another parameter that underwent experimentation was the size of the labelled target examples. The original model uses 50, but testing can be performed to determine if significant accuracy can be achieved from small increases in the size of this data, or whether reductions in the size of the data can still yield effective, accurate results. Experimentation was done on labelled samples of size 10, 25, 50, 75, 100, and 250 to compare the accuracy of the different labelled samples to determine whether changes can be made to the original model to improve either the accuracy or efficiency.

Number of tuning	average and list of accuracies
10	0.6557
25	0.6683
50	0.6448
75	0.6981
100	0.6530
250	0.6447

Table 3: Different labelled target domain sizes

This information is just a summary of the data from the mapping from book review labelled polarities to DVD review predictions from the indicated size of labelled target domain data. The overall trends in the results from this scenario and the other three show not a significant increase or decrease in accuracy based on the labelled sample size, with most peaking at 50 or 75 and the dropping off at sample sizes 100 and over. This doesn't make sense at first, but considering the increase in labelled sample size means smaller unlabeled and validation sets, a large increase in the labelled size would indeed hurt the accuracy of the model. Overall, it appears that labelled samples of even less than 50 can still achieve desired results, so in domains without much labelled data can still be tested relatively easily and effectively using the transferability of this model.

Experimentation of the parameters of the model determined that much of the original model was indeed correctly parameterized based on testing 3 different characteristics of the models, but it was revealed that the model can still effectively predict sentiment polarity with even less than 50 labelled samples. Implementation of this model verified that broad sentiment analysis of a wide variety of data formats is possible via a transferable sentiment analysis model trained on a large labelled dataset, which can then be adapted to various other datasets, whether or not there is much labelled data available, a huge step towards progress in natural language processing, and specifically sentiment analysis of a largely unlabeled dataset.

Plots created from the training process provide insight on

the distribution of source and target data in the hidden space of different representations (Figures 1-3).

3.4. Future Expansion of Model

Now that the model has successfully been implemented using the Amazon dataset, testing can be expanded to other datasets, both with and without unlabelled data. The first experiment would be to refactor the data loading module to be effective beyond the Amazon dataset to apply the model to other large labelled sentiment analysis datasets, such as the IMDB dataset of highly polar movie reviews. It would be interesting to see how this model performs on a more polarized dataset, but it would seem to have potential to be very high performing, especially considering the IMDB dataset is very large with enough labelled data as needed. This dataset could also be used as the source domain and applied to the Amazon target domain to see which performs more effectively.

Another application of the model would be to apply it to datasets lacking much labelled data, especially considering that it is these situations that this model is geared for, given its emphasis on transferability. This data could come in any form - online reviews of restaurants, hotels, apps, and other products, or even datasets created from scratch. The model really allows for a lot of flexibility in terms of the domains that it can be applied to, as long as the data intake is refactored to work with the data it is being applied to.

4. Conclusion

This implementation focused on development of a model based on a paper describing a cross-domain sentiment analysis model (Peng et al., 2018) capable of creating a base model from a large labelled dataset with the potential to transfer that model to a dataset with much less labelled data after necessary adaptation measures. This model proved to be effective in its ability to analyze the polarity of text of various domains outside the original domain the model was created from. Further experimentation with the model revealed that effective cross-domain analysis can be done with even less labelled data from the target domain than originally proposed by the paper. This opens that door for widespread effective sentiment analysis with or without large labelled datasets within the domain of natural language processing.

References

Büyüköz, B., Hürriyetoğlu, A., and Özgür, A. Analyzing elmo and distilbert on socio-political news classification. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*, pp. 9–18, 2020. URL https://www.aclweb.org/anthology/2020.aespen-1.4.pdf.

Figure 1. Figure 1: Domain Invariant representation mapping book reviews to DVDs (red points are source examples and blue are target examples)

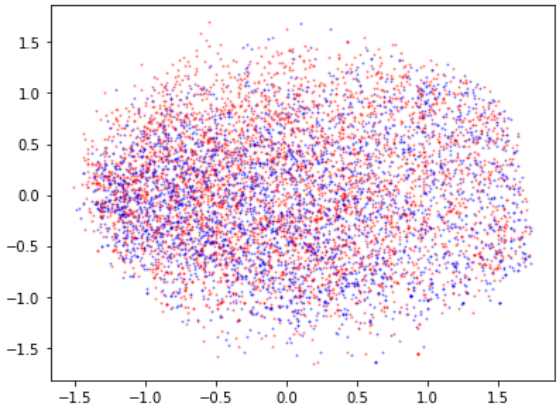


Figure 2. Figure 2: Source-Only representation mapping book reviews to DVDs (red points are source examples and blue are target examples)

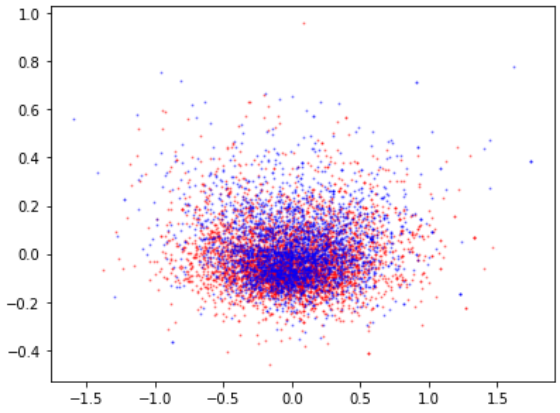
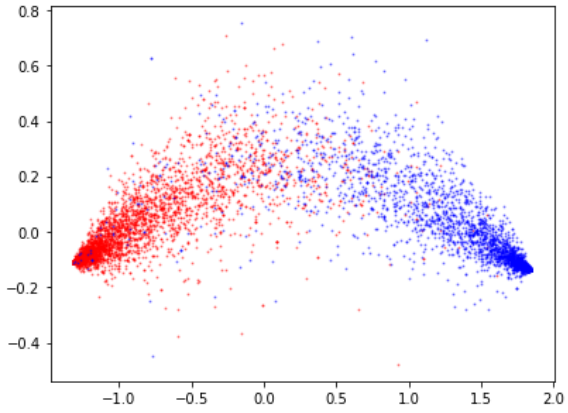


Figure 3. Figure 3: Target Specific representation mapping book reviews to DVDs (red points are source examples and blue are target examples)



Du, C., Sun, H., Wang, J., Qi, Q., and Liao, J. Adversarial and domain-aware bert for cross-domain sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4019–4028, 2020. URL https://www.aclweb.org/anthology/2020.acl-main.370.pdf.

Peng, M., Zhang, Q., Jiang, Y.-g., and Huang, X. Cross-domain sentiment classification with target domain specific information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2505–2513, 2018. URL https://www.aclweb.org/anthology/P18-1233.pdf.