Selecting Dissimilar Training Data for Cross-Domain Sentiment Analysis

Chrisanna Cornish
`ccor@itu.dk`

Danielle Dequin
`ddeq@itu.dk`

Sabrina Pereira
`sabf@itu.dk`

Abstract

Sentiment analysis is a domain specific task which requires a large amount of labelled data to train a high performing model. Training large models are compute intensive and labelling data is expensive. Therefore, it can be costly to develop a model to predict sentiment in a new domain. Existing models for cross-domain sentiment analysis select data from the source domain based on similarity to the target domain. We evaluate an approach to train a model using the least amount of training data possible by selecting the most dissimilar samples from the target domain. Additionally, we investigate the ideal threshold of data required for the model to perform as well on the target domain as it did on the source domain.

1 Introduction

In Natural Language Processing (NLP) most tasks such as sentiment analysis are domain-dependent, and as such require retraining a model for a new domain. Training a model in NLP can be both computationally and time intensive, and it is not always possible to get annotated data. Therefore, a lot of research in cross-domain sentiment analysis focuses on different ways to adapt a model to a new, low-resource target domain.

Current solutions tackle this problem in a variety of ways; from pre-training methods that tokenize the data in a specific way (Zhou et al., 2020) to post-training methods that allow the model to become domain-aware (Du et al., 2020). However, some of these methods will not apply to low-resource domains or a situation where there is no sufficient model available. In addition, as far as we know, there are no methods that cherry-pick training samples from the target domain in order to bridge the gap that exists when the domains are dissimilar.

In this project, we investigate the following research questions: *What is the amount of data needed to fine-tune a pre-trained model in cross-domain sentiment analysis? Can the amount of data be minimized by selecting dissimilar samples from the target domain?* Therefore, we tested increasing amounts of training data to see how it impacts model performance. We ran experiments with two target domains; one that is more similar to the source domain and another that less similar to investigate how the amount of training data needed varies based on source and target domain similarity.

For the baseline set of experiments, training samples were chosen at random from the target domain. In the next set of experiments, training samples were chosen from the target domain based on low cosine similarity score to the source domain.

2 Related Work

To our knowledge, there is no study that has examined the amount of data required for fine-tuning a pre-trained model in cross-domain sentiment analysis, with the training samples selected from the target domain based on dissimilarity. However, there are quite a few previous works that investigate the problem of cross-domain sentiment analysis.

Blitzer et al. (2007) uses a Structural Correspondence Learning (SCL) which attempts to learn the co-occurrence between features from the two domains. This method is used to select training data that matches a new domain, but does not have a solution when the source and target domain are dissimilar.

Pan et al. (2010) uses Structured Feature Alignment (SFA) which uses some domain-independent words as a bridge between domains. This method is specifically useful when the target domain is unlabelled, but is not optimal when the domain is low-resource.

We present an approach of selecting the training data from a target domain based on how dissimilar they are to the source domain. This method would require less training data and therefore applies well to low-resource domains.

3 Method

3.1 Data Description

The data used in this project are Amazon reviews (Ni, 2018), and we consider product categories as domains. We selected reviews from the product category, 'music' as our source domain. And the two target domains are 'video games' (games) and 'arts, crafts, and sewing' (sewing) reviews. Table 1 includes statistics on the datasets.

Table 1: Dataset statistics

	# samples	% positive
music	120.000	97%
sewing	339.610	93%
games	364.933	86%

The games data was chosen because according to KL-divergence, it is a more similar dataset to the source domain, while the sewing dataset was chosen because it is less similar.

3.2 Data Preparation and Processing

Instead of using only the 'reviewText' for training, we concatenated it with the 'summary' field to generate our samples. An example review from the music data is shown in Listing 1.

To generate the ground truth labels, the star ratings were used from the field 'overall'. To avoid ambiguity, 3-star reviews were removed, 1 or 2-star reviews were labelled as negative and 4 and 5-star reviews as positive.

The provided music data had 100.000, 10.000, and 10.000 samples in the train, development (dev) and test sets respectively. After pre-processing, the target domain data were split into train, dev and test sets. The splits were stratified to maintain the original label distribution. 10.000 samples were initially split off for the test set, and the remaining data was split 80/20 for train and dev sets.

Measuring Domain Similarity We used Kullback-Leibler (KL) divergence to measure the similarity between source and target domains in order to see if the proposed method varies depending on domain similarity. KL-divergence is a classical measure of 'distance' between two probability distributions, and is defined as:

```
{
    "overall":5.0,
    "vote":"3",
    "verified":true,
    "reviewTime":"06 3, 2013",
    "reviewerID":"A2TYZ821XXK2YZ",
    "asin":"3426958910",
    "style":{
        "Format:":" Audio CD"
    },
    "reviewerName":"Garrett",
    "reviewText":"This is awesome to listen to, A must-have for all Slayer fans..sadly needed to be a triple disc set..They have so many hits!!",
    "summary":"Slayer Rules!",
    "unixReviewTime":1370217600
}
```

Listing 1: Music review example

$$D_{KL}(q||r) = \sum_y q(y) log \frac{q(y)}{r(y)}$$

The reviews were tokenized with all words lower-cased, with stop words and punctuation removed. A combined vocabulary list was created of all tokens in the two domains being compared. This vocabulary list was used to get a count of each token, per domain. We used Laplace smoothing with a smoothing factor of $0,01$. This token frequency was converted into token probability by dividing by the total count of tokens in that domain. These two probability lists were then fed into a KL-divergence function.

The results of the KL-divergence between the target domains are in Table 2, showing that the games data are more similar to the music data than the sewing data are.

Table 2: KL divergence between target and source domains

Target Domain	KL-divergence
Sewing	2,01
Games	1,11

Measuring Training Data Similarity We used cosine similarity to compare and select dissimilar training data from the target domains. Cosine Similarity is a measure of similarity between two vectors obtained from the cosine angle multiplication value of two vectors being compared (Lahitani et al., 2016). It is given by:

$$cos(a,b) = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \cdot ||\vec{b}||} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}}$$

The smaller the angle between the word vectors, the more similar they are. A cosine similarity of 0 is most dissimilar, and 1 is most similar.

To compute the cosine similarity between source and target domains, all reviews were pre-processed which included; tokenizing each review, removing stop words and lower casing all words. We created a corpus, in this case a bag of words, for the domain. A tf-idf model of the corpus was then created to take into account word frequency and decreases the weight for commonly used words.

Each review in the target domains was compared to all reviews in the source domain to calculate their individual cosine similarity. We then calculate the average score per review. Reviews with the lowest average cosine similarity were considered the most dissimilar and selected as the training set for the experiments.

3.3 Evaluation

The chosen metric to measure performance was f1 score because the datasets are very unbalanced. With the imbalance, accuracy would be high even if all samples were predicted as positive. F1 score is the harmonic mean of precision and recall, which gives much more weight to low values. Therefore, a high f1 score would only be achieved if both precision and recall are high (Géron, 2019).

4 Experiments and Results

Following the implementation by Tran (Tran, 2020), we used a pre-trained BERT language model to get the embeddings for each review, which were then fed into a one-layer feed forward neural network using ReLU as an activation function. The neural network outputs the probability for each of the two classes (positive or negative) and predicts the class with the highest probability. The model was then fine-tuned on the music training data. This formed our baseline model for further experiments.

Experiment 1: Baseline

As an experiment baseline, we tested how the baseline model trained on only music data performs on each target domain without additional fine-tuning. The results of the baseline experiments can be seen in Table 3, including how the model performs in the music domain.

Table 3: Experiment 1: Baseline

Experiment	Domain	f1 Score
Base	Music	0,938
Base	Sewing	0,850
Base	Games	0,778

The baseline model is able to predict better in the sewing data than the music data. We had expected better performance in the games domain, given that the music and games domains were more similar. However, the pattern seems to coincide with the label distributions in each domain, as shown in Table 1.

Experiment 2: Random Data

This experiment consisted of training the model four times on each target domain, with increasing training set size. We selected sets of 10, 100, 1.000, and 10.000 random reviews from each domain. We looked into the distribution of labels to see how this compared to the distribution in the original dataset. The percentage of positive reviews in each training set are shown in Table 4.

Table 4: % Positive in Random Training Samples

Sample size	Sewing	Games
10	80%	100%
100	92%	90%
1.000	94%	89%
10.000	94%	87%

The baseline model was fine-tuned with each subset of data separately. This fine-tuned model was then used to make predictions on the test set. We evaluated performance using the weighted f1 score. Table 5 shows the results of these experiments.

Note that an f1 score of 0,820 on games and 0,913 on sewing is achieved by predicting all reviews to be positive.

Table 5: Experiment 2: Random Data

Experiment	Domain	f1 Score
Base + 10	Sewing	0,929
Base + 100	Sewing	0,948
Base + 1.000	Sewing	0,913
Base + 10.000	Sewing	0,965
Base + 10	Games	0,862
Base + 100	Games	0,925
Base + 1.000	Games	0,820
Base + 10.000	Games	0,820

Experiment 3: Selective Data

In this experiment, we used the same set sizes, but selected training data from the target domain based on dissimilarity to the source domain.

Again we looked into the distribution of labels in these sets to compare against the distribution in the original dataset. The percentage of positive reviews in each training set are shown in Table 6. Coincidentally, the percentages are the same across both target sets.

Table 6: % Positive in Selective Training Samples

Sample size	Sewing	Games
10	100%	100%
100	95%	95%
1.000	89%	89%
10.000	89%	89%

The base model is again fine-tuned in each experiment, and then used to predict sentiment on the test set. The results are shown in Table 7.

Table 7: Experiment 3: Selective Data

Experiment	Domain	f1 Score
Base + 10	Sewing	0,967
Base + 100	Sewing	0,913
Base + 1.000	Sewing	0,969
Base + 10.000	Sewing	0,913
Base + 10	Games	0,868
Base + 100	Games	0,919
Base + 1.000	Games	0,820
Base + 10.000	Games	0,820

With the percentages of positive labels being the same in both target domains, it is easier to compare the results and see that the model performs better overall when predicting on the sewing data.

Experiment 4: Balanced Data

Both the games and sewing datasets were heavily unbalanced. This meant the model could perform well, even using the f1 score, by labelling everything as the positive class. An additional experiment was then carried out in order to attempt to address this by undersampling from the positive class so that an equal number of positive and negative samples were selected in the random sample. The results are shown in Table 8.

Table 8: Experiment 4: Balanced Random Data

Experiment	Domain	f1 Score
Base + 10	Sewing	0,876
Base + 100	Sewing	0,967
Base + 1.000	Sewing	0,933
Base + 10.000	Sewing	0,950
Base + 10	Games	0,826
Base + 100	Games	0,837
Base + 1.000	Games	0,876
Base + 10.000	Games	0,820

This was more successful at not predicting all reviews as positive. However, the number of incorrect predictions increased, particularly the false negatives.

5 Analysis

We investigated the amount of data needed to fine-tune a pre-trained model in cross-domain sentiment analysis, however it is unclear if there is an ideal threshold of data required. The results do not show an ever-increasing performance that correlates with the amount of data, nor does the performance peak at a given threshold.

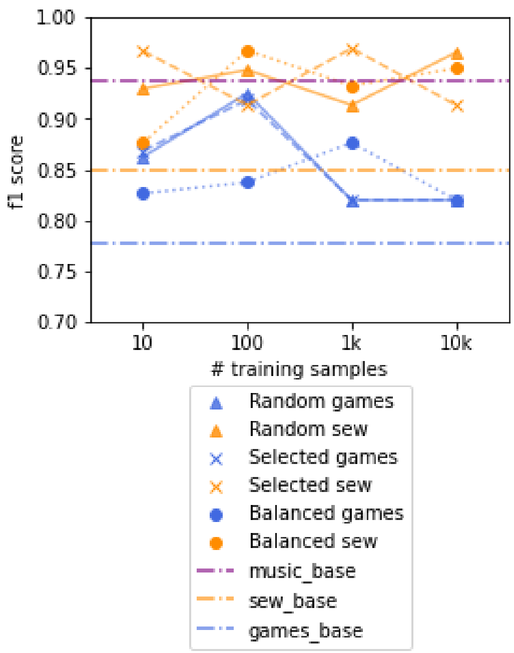
We also investigated if the amount of data can be minimized by selecting dissimilar samples from the target domain, however there does not seem to be a clear advantage in using this method compared to random selection.

The model performed better on the more dissimilar dataset (sewing) than it did on more similar one (games). This is the case in almost all experiments. We believe that this may be due to the difference in label distributions between the sewing and games datasets. As is shown in the results in Tables 5, 7, and 8, the lowest results are achieved with all reviews predicted positive. The lowest score in the sewing data is about 10% higher than the lowest score in games data, which correlates to the 10%

difference in positive labels.

Figure 1 shows the f1 scores across all experiments. The horizontal lines are the baseline scores before any fine-tuning was performed. Performance measures regarding the sewing data are in yellow, while the games results are plotted in blue. How the baseline model performed in its own domain, music, was the performance goal, shown in purple.

Figure 1: f1 scores of experiments 1-4



The games experiments produced very similar results whether the model was fine-tuned on selected or random data, although it tended to differ in which reviews mistakes were made on. It is interesting that results improved with smaller samples but when larger samples were given, it predicted all samples as positive.

Balancing the classes during random data selection had mixed results on performance. It improved performance in 1 out of 4 experiments for the games dataset and in 2 out of 4 experiments for the sewing dataset.

We believe some of the ambiguity in the results is due to confounding factors. For example, when manually inspecting the selected dissimilar reviews, we noticed that the most dissimilar samples from the games data were in Spanish, whereas most of the dissimilar reviews from the sewing dataset were in English. The baseline model was trained on the music data, which seems to be mostly in English. Additionally, there were many duplicate samples in the datasets. For example, there were many reviews containing the text ‘excelente excelente’ in the games data. The repetition in the training data decreased variety in the samples the model could have learnt from. The label distribution, language and number of repetitions varies in both target domains, which impacts the interpretability of the results.

6 Conclusions

Regarding our research question, we did not find a clear result indicating the ideal threshold of training data required. We expected increased performance as training data increased. However, we did not find that pattern in our results. This could be due to the quality of the training data provided to the model and other confounding factors. These confounding factors include the language differences between the two target datasets, variation in repeated training samples, and the heavy imbalance in label distributions. Additionally, the variation in percentage of each class varied between all experiments.

Regarding the method to select training data from a target domain based on dissimilarity, we found that random selection of training data might be a better choice, specially when compute is a scarce resource. Both methods achieve similar results, however the time required to process the data for dissimilarity make this a less favourable choice. However, it is difficult to come to a hard conclusion here due to the previously stated confounding factors. Further, more meticulous, research could be done to investigate the ideal threshold of training data needed, and if data can be selected from a target domain based on dissimilarity.

References

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447.

Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, and Jianxin Liao. 2020. Adversarial and domain-aware bert for cross-domain sentiment analysis. In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pages 4019–4028.

Aurélien Géron. 2019. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* ” O’Reilly Media, Inc.”.

Alfirna Rizqi Lahitani, Adhistya Erna Permanasari, and Noor Akhmad Setiawan. 2016. Cosine similarity to determine similarity measure: Study case in online essay assessment. In *2016 4th International Conference on Cyber and IT Service Management*, pages 1–6. IEEE.

Jianmo Ni. 2018. Amazon review data.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760.

Chris Tran. 2020. Tutorial: Fine-tuning BERT for Sentiment Analysis - by Skim AI. [Online; accessed 24. May 2022].

Jie Zhou, Junfeng Tian, Rui Wang, Yuanbin Wu, Wenming Xiao, and Liang He. 2020. Sentix: A sentiment-aware pre-trained model for cross-domain sentiment analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 568–579.

A Group contributions

All members of the group contributed to various aspects of the project.

B Results of Early Phases of the Project

B.1 Results of Phase 1

Our baseline model predicted on the test set with 0,938 f1 score.

B.2 Results of Phase 2

The average performance of our baseline model on the hard cases was 0,674.

C Link to GitHub Repository

The GitHub repository contains all the code and documentation necessary to reproduce the results of this project:

https://github.itu.dk/ddeq/2yp_sentiment_analysis.git