Dissimilar VS Randomly Selected 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. For organizations without those resources it could be costly to develop a model to predict sentiment in a new domain. Existing models for cross-domain sentiment analysis choose data from the source domain based on similarity to the target domain. In this paper, we evaluate a different approach where we use a pre-trained BERT model and select training data from the target domain based on dissimilarity to the source domain. Separate experiments were run to compare performance when data is selected randomly from the target domain. Additionally, we explore the different amounts of training data in order to see if there is an ideal threshold in order for the model to perform equally well cross-domain as it did it it's original domain. We find that there is little improvement, depending on the amount of data used in training, when selecting the data based on dissimilarity to the source domain compared to randomly selecting training samples from the target domain.

1 Introduction

In Natural Language Processing (NLP) the task of sentiment analysis is largely dependent on the domain, therefore requiring a new model per domain. Training a model in NLP can be both computationally and time intensive, and many domains lack substantial resources for training. 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 to post-training methods that allow the model to become domain-aware. However, some of these methods will not apply to low-resource domains

In this project, we investigate the minimum amount of training data required to fine-tune a pre-trained BERT model on a new target domain to match its performance on the source domain. Therefore, we tested increasing amounts of training data to see how it impacts model performance. Experiments were run in parallel 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.

Additionally, we investigated how the models performed when the training data was selected using two methods. In one set of experiments training samples where chosen at random from the target domain. In the other set of experiments training samples were chosen based on dissimilarity to the target domain.

2 Related Work

To our knowledge, there is no study that has examined specifically the different amounts of data required for fine-tuning a pre-trained model in cross-domain sentiment analysis, while comparing the differences when data is randomly versus specifically-selected. 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 came from Amazon reviews (Ni, 2018). In this setting, product categories are considered the domains. For this experiment we selected 'music' reviews as our source domain and the target domains are reviews from the categories 'video games' (games) and 'arts, crafts, and sewing' (sewing).

3.2 Data Preparation and Processing

To label the data, the star rating of the review was used, which range from 1 to 5 stars. A rating of 3 was considered neutral and those reviews were removed. Reviews with ratings of 1 and 2 stars were considered negative and given the label '0', and ratings of 4 and 5 were considered positive and given the label '1'.

Each review contained a summary text, and review text. These two fields were concatenated into a column called 'review'. The concatenated text was used to train and evaluate the model. The same process was used for predictions.

After pre-processing the data was split into train, development (dev) and test sets which were stratified to maintain the original label distribution. 10.000 samples were split off first for the test set, and the remaining split 80/20 for train and dev sets. After splitting, the datasets were saved as .csv files containing only the 'review' and 'label' columns.

Measuring Domain Similarity Kullback-Leibler (KL) divergence was used to measure the similarity between the domains. KL-divergence is a classical measure of 'distance' between two probability distributions, and is defined as:

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

It is a non-negative, additive, asymmetric measure, and equals 0 if the two distributions are identical (Plank and Van Noord, 2011).

To prepare the data, the reviews were tokenized using nltk's word_tokenize. Each tokenized review had stop words and punctuation removed, and all words lower-cased. A list was created of all tokens in the combined vocabulary of the two domains being compared.

Then a count of all tokens was done, per domain, based on this vocabulary list. Since some tokens will not exist in both lists $0, 01$ was added to those token counts to avoid zero division later on. This token count was converted into token probability for each domain, based on the count of all tokens in that domain. These two probability lists were then fed into a KL-divergence function.

The KL-divergence between the music and games data sets is 1,11 whereas the score between music and sewing is 2,01. This indicates that the music and games data are more similar than the music and sewing data.

Table 1: KL divergence between target and source domains

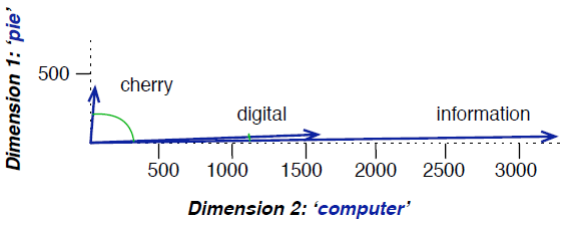
Target Domain	KL-divergence
Sewing	2,01
Games	1,11

Measuring Training Data Similarity Cosine similarity was used to compare and select dissimilar training data. 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.

Figure 1: A graphical example of cosine similarity for the words cherry, digital, and information in two-dimensional space (Jurafsky, 2000).



To compute cosine similarity, the music dataset was first processed. This processing included; each review was tokenized using nltk's word_tokenize, with stop words removed and all words lower-cased. The gensim library was used to create a dictionary mapping each token to a number. This dictionary was then used to create a corpus, in this case a bag of words, for the domain.

A tf-idf model of the corpus was created using gensim's TfidfModel. The tf-idf model was then applied to the corpus, which was fed into gensim.similarities.Similarity to create a cosine similarity measure object. We chose to use tf-idf to take into account word frequency and decreases the weight for commonly used words.

The comparison domains went through the same process as the music data, and a bag of words created per review. Tf-idf was applied, and the processed review was fed into the similarity object to get the cosine similarity of the review against all reviews in the base corpus. The average of this was computed, and appended as a column to the data as the average cosine similarity per review.

The data were sorted by dissimilarity. The top 10, 100, 1.000, and 10.000 most dissimilar reviews were then saved as separate .csv files for both the sewing and games domains.

When manually inspecting the dissimilar reviews, the most dissimilar samples from the games dataset were in Spanish, whereas the dissimilar reviews from the sewing dataset were in English. Additionally there seem to be a lot of duplicate reviews. For example, there were many reviews containing the text 'excelente excelente' in the games data. The repetition in the training data is therefore decreasing the variety that the model will learn from.

3.3 Experiment Setup

Baseline model A pre-trained BERT language model (Tran) was used to turn tokenized reviews into embeddings which are then fed into a simple one layer feed forward neural net using ReLU as an activation function. The neural net outputs the probability for each of the two classes (positive or negative) and predicts the class with the highest probability.

The baseline model was then trained on 100.000 samples of digital music reviews for the phase 1 task. The model uses cross entropy to compute the loss. This formed our base model for the further experiments.

Performance Metrics The chosen metric to measure performance was f1 score, which is the harmonic mean of precision and recall. The harmonic mean gives much more weight to low values, and therefore a high f1 score would only be achieved if both precision and recall are high (Géron, 2019).

As the datasets are very unbalanced, 94% of the reviews are positive in the sewing dataset and 88% positive in the games data, so f1 score is a better measure than accuracy which would be high even if all samples were predicted as positive.

Experiment 1: Baseline As a baseline, we tested how the base model performs on each target domain without fine-tuning it on any additional data.

Experiment 2: Random Incremental Data The experiment consisted of training the model four times for each target domain, each time increasing the quantity of training samples taken randomly from the target domain. The pandas dataframe sample method was used, with the random_state parameter set to 42.

Sets of sizes 10, 100, 1.000, and 10.000 reviews were randomly selected from each domain. The percentages of positive reviews in each training set are shown in Tables 2 and 3.

Table 2: Games training set

Size	% Positive
10	100%
100	90%
1.000	89%
10.000	87%

Table 3: Sewing training set

Size	% Positive
10	80%
100	92%
1.000	94%
10.000	94%

The base model was fine-tuned with each subset of data separately, and then predictions were made on the test set and the performance evaluated using f1 score.

Experiment 3: Selective Incremental Data The experiments were repeated with the same set sizes,

but instead of selecting random data from the target domains, they were selected based on dissimilarity to the source domain.

The percentage of positive reviews in this selectively picked data were 100%, 95%, 89% and 89% for 10, 100, 1.000, and 10.000 respectively in both sewing and games data.

The base model is again fine-tuned in each experiment, and then used to predict the sentiment of the test set.

The schedule of experiments and evaluation can be seen in table 6.

4 Results

The results of the baseline experiments can be seen in Table 4. Without any fine-tuning the model has an f1 score of 0,850 in the sewing data and 0,778 in the games data.

Table 4: Experiment 1: Baseline

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

The results of experiment 2 where the training data are randomly selected are shown in Table 5.

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

The results of experiment 3 where the target domain training data are selected based on dissimilarity to the source domain are shown in Table 6.

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

Both the games and sewing datasets were heavily unbalanced. This meant the model could perform well, even using the f1 metric, by labelling everything as the positive class. Experiment 4 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 7.

Table 6: 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

Table 7: 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 + 10.00	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

The model performed better on more dissimilar data (sewing) than it did on more similar data (games). This is the case in almost all experiments, whether the training data from the target domain was randomly selected or selected based on dissimilarity to the source domain.

Figure 2 shows the f1 scores across all initial 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.

Figure 2: f1 scores across initial experiments

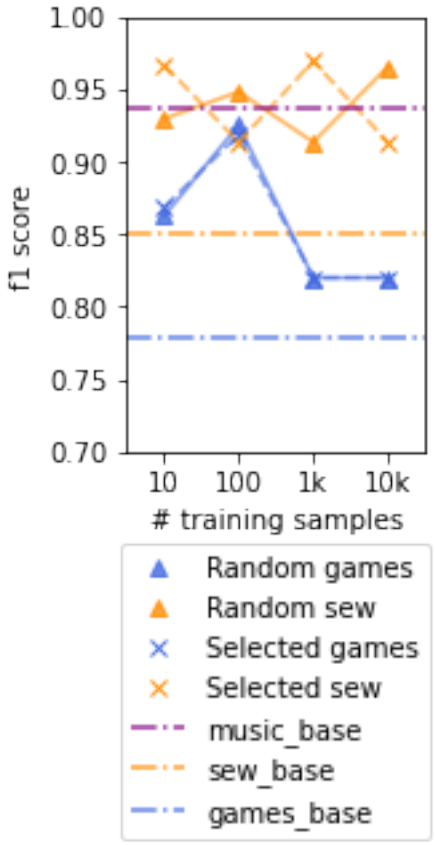
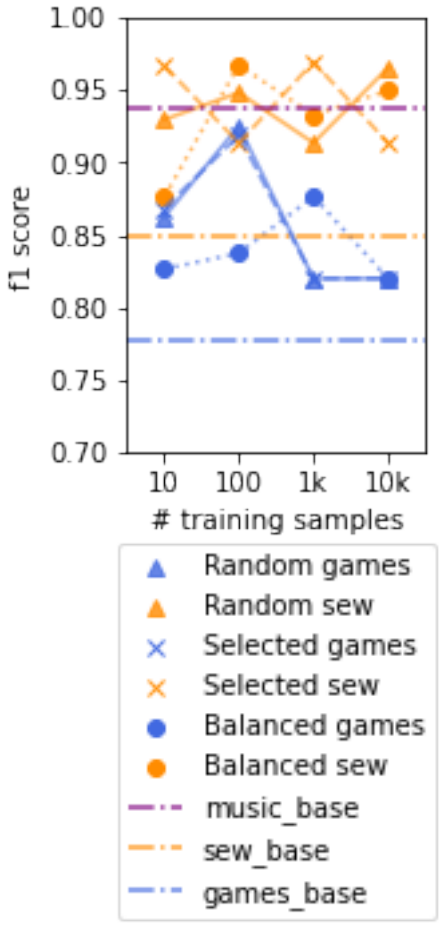


Figure 3: f1 scores with balanced trial group



The instance where the model performs better in the games domain versus the sewing domain, where the f1 score for sewing drops to 0,913. This occurred when the 100 sewing data samples were selected based on dissimilarity.

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

Figure 3 shows the f1 scores for all the retraining experiments.

6 Conclusions

We expected increased performance with each addition of training data, however, we found that performance does not necessarily increase. This could be due to the quality of the training data provided to the model and other confounding factors. One confounding factor can be the language differences between the two target datasets. Another factor can be that the datasets in general were heavily unbalanced. Additionally, the variation in percentage of each class varied among all experiments.

We found that random selection of training data from our source domains might be a better choice than selecting training data based on cosine similarity, 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.

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.

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.".

Dan Jurafsky. 2000. *Speech & language processing*. Pearson Education India.

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.

Barbara Plank and Gertjan Van Noord. 2011. Effective measures of domain similarity for parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1566–1576.

Chris Tran. Tutorial: Fine tuning bert for sentiment analysis.

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 following is a link to the GitHub repository that contains all of the code and documentation necessary to reproduce the results of this project.

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