Divergence-Based Domain Transferability for Zero-Shot Classification

Alexander Pugantsov and **Richard McCreadie**
School of Computing Science, University of Glasgow, UK

Abstract

Transferring learned patterns from pretrained neural language models has been shown to significantly improve effectiveness across a variety of language-based tasks, meanwhile further tuning on intermediate tasks has been demonstrated to provide additional performance benefits, provided the intermediate task is sufficiently related to the target task. However, how to identify related tasks is an open problem, and brute-force searching effective task combinations is prohibitively expensive. Hence, the question arises, *are we able to improve the effectiveness and efficiency of tasks with no training examples through selective fine-tuning?* In this paper, we explore statistical measures that approximate the divergence between domain representations as a means to estimate whether tuning using one task pair will exhibit performance benefits over tuning another. This estimation can then be used to reduce the number of task pairs that need to be tested by eliminating pairs that are unlikely to provide benefits. Through experimentation over 58 tasks and over 6,600 task pair combinations, we demonstrate that statistical measures can distinguish effective task pairs, and the resulting estimates can reduce end-to-end runtime by up to 40%.

1 Introduction

As the accuracy of neural models continues to increase, so does the computational cost of training and storing them. One approach of mitigating such cost is through using pretrained models to enhance performance on a downstream task, a paradigm commonly referred to as *transfer learning*. However, when and why transfer learning works is not concretely understood. Traditionally, selecting the best settings, i.e. tasks and hyperparameters, for transfer often involves an extensive trial-and-error process over many combinations and can quickly make the prospect of applying transfer learning undesirable. As such, it would be valuable to estimate whether a task pair combination will be effective pre-training, i.e. estimate the *transferability* of a source task to a target task.

The most optimal transferability metric would be resource-efficient, such that it is capable of accurately predicting the final performance of the model whilst minimising the amount of processing required to compute it. To this end, several works (Van Asch and Daelemans, 2010; Ruder and Plank, 2017; Ramesh Kashyap et al., 2021) have focused on estimating transferability prior to fine-tuning, using statistical measures of divergence between the underlying feature spaces of model pairs. Domain divergence measures are used to produce a notion of distance between pairs of domains by comparing their representations and have seen significant usage in works which investigate the correlation between their estimations and performance change (Van Asch and Daelemans, 2010; Ramesh Kashyap et al., 2021).

Subsequent transfer learning works have also demonstrated that competitive model performance can be achieved on some target tasks even if no training samples for that task are available, an approach known as *zero-data/shot learning* (Larochelle et al., 2008). In this work, we investigate the effectiveness of domain divergence measures in estimating the performance of zero-shot classification models, wherein models further tuned on one source task are used to directly predict on the test set of a target task without any target training samples. Specifically, we leverage the information captured by these measures as features to an auxiliary learner, whose outputs are used to rank the most effective source model for transfer to a given target task. Through the analysis of 58 sentiment classification domains, we: (1) perform a correlation analysis between each independent measure and each source-target, macro-averaged F_1-score performance output; (2) and, for each target task, we train a series of auxiliary regression models to predict their projected performance;

Findings of the Association for Computational Linguistics: EACL 2023, pages 1649–1654

(3) we then convert these into rankings of source–target pairs and evaluate the capability of our learners to find the best source model for each given target domain.

2 Experiment Setup

Measures: Ramesh Kashyap et al. (2021) provide categories of divergence measures; two of which we use in our work: *Geometric* measures which calculate distances between continuous representations such as word embeddings and *Information-theoretic* measures which capture the distance between representations such as frequency-based distributions over co-occurring n-grams. We do not report higher-order measures as in the aforementioned work, but instead report *moments*-based features, which better describe the characteristics of our individual term distributions—namely the mean, variance, skewness, and kurtosis of our distributions—as features to our learner. Following prior work (Tsvetkov et al., 2016; Ruder and Plank, 2017), we further complement the above measures by making use of several metrics that capture diversity and prototypicality such as entropy-based features; in our work, these measures are used with probability distributions, and are, as such, categorised here as information-theoretic. Specifically, we use the following metrics:

- **Geometric**: Cosine distance, l_1- (or Manhattan dist.) and l_2-norm (or Euclidean dist.).
- **Information-theoretic**: Rényi and Jensen-Shannon divergences (Wong and You, 1985; Rényi et al., 1961), Bhattacharyya Coeff. (Bhattacharyya, 1943), Wasserstein distance (Kantorovich, 1960), Entropy and Rényi Entropy (Shannon, 1948; Rényi et al., 1961), Simpson's Index (Simpson, 1949).
- **Moments-based**: Mean, variance, skewness, and kurtosis (σ^n where $n \in [1..4]$).

Representations: To compute the above metrics, we use two different representations from prior work by Ruder and Plank (2017), specifically 1) discrete probabilities of the most common terms across domains, using a fixed-size vocabulary V, where $|V| = 10,000$; and 2) a summation over probability-weighted term embeddings in each document, averaged to produce a single vector:

(1) **Term Distributions (TD)** (Plank and van Noord, 2011): $t \in \mathbb{R}^{|V|}$ where t_i is the probability of the i-th word in the vocabulary V.

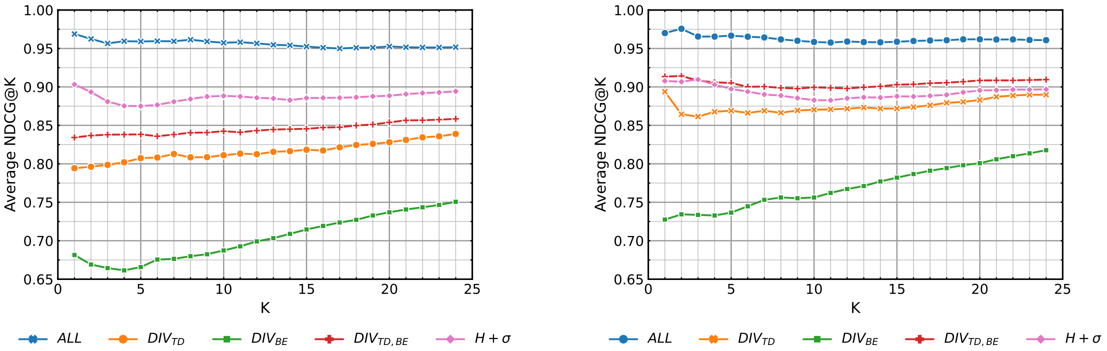
(2) **BERT Embeddings (BE)** (Devlin et al., 2018): $\frac{1}{n}\sum_i v_{w_i}\sqrt{\frac{a}{p(w_i)}}$ where n is the number of words with embeddings in the document, v_{w_i} is the pretrained embedding of the i-th term, $p(w_i)$ its probability, and a is a smoothing factor used to discount frequent probabilities. Following guidelines by Ruder and Plank (2017), we use this representation with geometric-based measures only, as embedding vectors can be negative.

Generally, since we are using these representations in a zero-shot setting, we compute divergences between the source-task training set (D_S) and the target-task test set (D_T). Entropy and moments-based measures are not used to estimate divergence between domains but used only to compute within-domain characteristics, i.e. on individual term distributions.

Datasets and Domains: We make use of two ratings prediction datasets with classes in the range 1-5 and, similarly to Zhang et al. (2015), reformulate the task as a binary sentiment classification task by merging the provided labels; 1-2: negative and 3-4: positive. We focus on similar, within-task (i.e. sentiment classification) datasets to (1) remove task variation as a variable, (2) and to highlight the effectiveness of using statistical measures to compute divergence between similar domains which may have very minute differences in semantics and other linguistic phenomena. The first is the *Amazon Product Reviews* dataset, using the review title and review content fields as features and divide the dataset by the product category labels. As a supplementary contribution to our work, we create the *Multi-Domain Yelp Business Reviews* dataset by extending the original *Reviews* and *Business* datasets provided by the *Yelp Dataset Challenge*, mapping top-level[1] categories of businesses to their respective reviews. After filtering out low-sample ($\leq 30,000$) domains, we have 42 and 16 domains for the Amazon and Yelp datasets, respectively.

Implementation Details: We use BERT$_{base}$ (Devlin et al., 2018) as our base model in the experiments. With both runtime- and storage-efficiency in mind, we make use of adapter modules (Pfeiffer et al., 2020) and train each of the domains as a source task adapter, leaving the rest of BERT's

[1]We determined which categories were top-level based on an article written by Yelp



(a) Average NDCG@K for $N_S = 1000$.

(b) Average NDCG@K for $N_S = 25000$.

Figure 1: F1@K averaged across tasks vs. Total Runtime@K of source-task adapters. Higher is better. Runtime is reported in hours.

parameters frozen. More implementation and hyperparameter details can be found in Appendix A.

We divide our experiments into two separate settings by source-task sample size, $N_S \in [1000, 25000]$. We train 116 source-task adapters (58 D_S $\times$ 2 N_S settings), and evaluate a total of 6,612 source-target combinations for analysis. For our auxiliary learner, we use an XGBoost (Chen and Guestrin, 2016) regression model. We split our training and test sets by the target task and train 2,900 regression models (for each of the 58 target domains, 2 sample sizes settings, 5 feature sets, and over 5 random seeds).

3 Experiments and Results

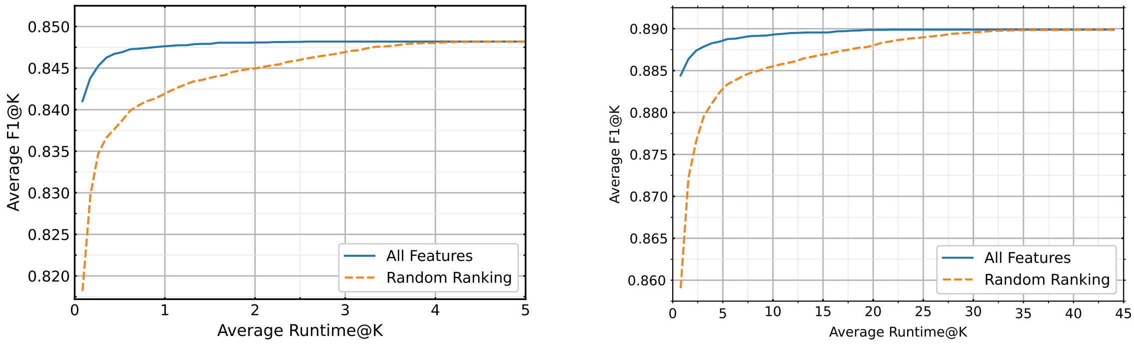
Category	Measure	Term Distributions		BERT Embeddings	
		1K	25K	1K	25K
Geometric	Cosine Dist.	-0.3683*	-0.4801*	-0.3078*	-0.5792*
	L_1 Dist.	-0.3699*	-0.6243*	-0.0792*	-0.4045*
	L_2 Dist.	-0.3345*	-0.3551*	-0.0923*	-0.4228*
Info. Theoretic	Rényi Div.	-0.4766*	-0.4273*		
	Jensen-Shannon Div.	-0.3726*	-0.5914*		
	Wasserstein Dist.	-0.2225*	-0.3266*		
	Bhattacharyya Coeff.	0.3700*	0.5743*		
	Entropy (D_S)	0.1838*	0.2275*		
	Entropy (D_T)	-0.1603*	0.0486*		
	Rényi Entropy (D_S)	-0.1836*	-0.2284*		
	Rényi Entropy (D_T)	0.1618*	-0.0503*		
	Simpson's Index (D_S)	0.0842*	0.1359*		
	Simpson's Index (D_T)	-0.3127*	-0.1442*		
Moments Based	σ^1 (D_S)	-0.1321*	-0.1792*		
	σ^1 (D_T)	-0.1245*	-0.2227*		
	σ^2 (D_S)	-0.1289*	-0.1523*		
	σ^2 (D_T)	-0.1749*	-0.2549*		
	σ^3 (D_S)	0.0106	0.0287		
	σ^3 (D_T)	-0.3823*	-0.2643*		
	σ^4 (D_S)	0.0006	0.0234		
	σ^4 (D_T)	-0.3491*	-0.2473*		

Table 1: Spearmans ρ correlations between each measure and source-target macro-averaged F_1-score performance. Asterisk denotes measure was statistically significant ($P \leq 0.05$).

To evaluate whether the aforementioned statistical measures are predictive of task pair transferability, we perform a correlation analysis between the source-target pairs within each domain, where we contrast the statistical measure (which provides information about D_S, D_T, or the differences between them) and the resultant performance (measured using macro-averaged F_1) when using D_S to tune a model for application on task T. Table 1 reports Spearman's Rho (ρ) across all sample size settings for each statistical measure. Higher correlations (distance from 0) indicate increasing predictiveness of the statistical measure of transferability.

Using the interpretation of Spearman's Rho (ρ) correlation coefficients by Dancey and Reidy (2007), we make the following observations: (1) Geometric measures exhibited a moderate–to-strong correlation for Term Distributions across both sample size settings, and strong correlations at $N_S = 25000$ for BERT Embeddings; (2) Between-domain Information-theoretic measures also showed moderate-to-strong performance correlations; (3) All entropy-based measures (aside from Simpson's Index for D_T) had a weak or negligible correlation with performance; (4) Out of all of the higher-order moments of Term Distributions, only the skewness and kurtosis of D_T (σ^3 and σ^4) seemed to have a moderate relationship at $N_S = 1000$, and, generally, the moments of D_T seemed to be more correlated than that of D_S.

Overall, divergence measures with both representations seemed to be more predictive of source-target performances than with entropy or moments-based metrics. However, since it is unlikely that each measure was independently capable of predicting performance, we trained a series of regression models for each target task, combining these measures. Specifically, we train an XGBoost (Chen and Guestrin, 2016) regression model (XGBRegressor) with each of the feature sets as our inputs, over five



(a) Average F1@K vs. Runtime@K for $N_S = 1000$.

(b) Average F1@K vs. Runtime@K for $N_S = 25000$.

Figure 2: F1@K averaged across tasks vs. Total Runtime@K of source-task adapters. Higher is better. Runtime is reported in hours.

random seeds, for each of the 58 target domains and 2 sample size settings, producing 2,900 models for evaluation.

Figure 1 shows the *Average NDCG@K* values for each of these feature sets. We average the NDCG@K values across each of the 58 domains, and again over each of the 5 seeds. For both models, we achieve the best quality ranking using all of the features (ALL). Moreover, using divergence measures with both sets of representations ($DIV_{TD,BE}$) achieved a better ranking than using them in isolation (DIV_{TD} or DIV_{BE}) for both settings. It is also interesting to note that the feature set containing only the entropy and moments-based ($H + \sigma$) values achieve better performance than that of those estimated via divergence measures when the source sample size is significantly limited, coinciding with patterns found in our correlation analysis (Table 1); it may be the case that these features are more discriminative in cases where divergence measures are not as expressive.

Finally, we evaluate the practical, downstream application of our regression models by considering how they may be used to reduce the search time in finding appropriate source models for transfer. For this experiment, we assume the user has a particular training budget K to train task pairs for transfer. The more task combinations that are tried, the more likely the user is to find a better-performing model for a particular task. We use our regression models to determine the order of task pairs to be tried, using the best feature set from our prior experiments (See Fig. 1). We compare with a random ordering of source-task models, which we average over five random seeds to reduce variance. Figure 2 shows the results of our experiments. For $N_S = 1000$, the best macro-averaged F_1 performance score over all tasks is 0.8482 which, with a grid search over all task combinations, would require 4.7 hours of training. With our approach, we can achieve a 44% reduction in training time from 4.7 to 2.6 hours to achieve the same performance. For $N_S = 25000$, we can achieve the maximum score of 0.8899 through a grid search of all source-target combinations at a cost of 42.4 hours of training time. With our approach, we can achieve the same score with only 24.9 hours of training or a 41% reduction in training time.

In determining the overall runtime of our approach, we factor in the computational cost associated with generating the features required to train our regression models. Our feature generation process consists of three stages: (1) the generation of term distributions and embedding representations, (2) the computation of statistical measures in Table 1, (3) and the execution of regression experiments using the ALL feature set. A total of 232 term distributions and an equivalent number of embedding representations (58 target domains each with separate training and test sets, in two different sample size settings) were generated. The generation of both sets of representations takes 5.7 minutes at $N_S = 1000$ and 45.9 minutes at $N_S = 25000$. The time taken to compute all statistical measures across both representations is 3 minutes at $N_S = 1000$ and 6.6 minutes at $N_S = 25000$. Finally, the time taken to run the regression experiments was 5.4 minutes in total. Despite the added computational cost, our approach has resulted in a substantial reduction in end-to-end runtime, boasting a 40% reduction at $N_S = 1000$ and a 39% reduction at $N_S = 25000$, demonstrating the efficiency of our approach and the value-add of predicting which task pairs are transferable beforehand.

4 Conclusions and Future Work

In this paper, we have shown that domain divergence measures and other statistical quantities are predictive of zero-shot transferability between tasks, and that this can be used to markedly reduce time when developing effective zero-shot models. Indeed, by predicting which source-target task pairs were likely transferable pre-tuning, we were able to reduce the end-to-end time taken to find the best source-target task pairs (trained on 1,000 source-task samples) by 40%. On the other hand, while we have demonstrated the value of using these metrics in performance estimation, there are a number of further directions worth investigating, namely: (1) examine the transferability across a wider range of domain and task types; (2) investigate more complex, higher-order measures such as those outlined by Ramesh Kashyap et al. (2021); (3) and to experiment with few-shot and other limited data settings.

Limitations

The most pronounced limitation in our work is the small variance in performance scores. As can be seen in Figure 2, the difference between the lower and maximum performances is small. The difference between the minimum and maximum average performance is 0.0305 and 0.0320 for $N_S = 1000$ and $N_S = 25000$, respectively. Even at the individual, source-target model level, the standard deviation of performance scores at each source-task sample size setting is 0.0363 and 0.0311. As such, the benefits of zero-shot transfer are not as apparent between these domains as they would be where the domains are more textually distinct. Nevertheless, we believe it is notable that statistical measures of domain divergence and the other metrics were sufficiently capable of discerning between more effective source-task pairs, even when the domains were similar, illustrating the promise of this approach.

References

Anil Bhattacharyya. 1943. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35:99–109.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754.

Christine P Dancey and John Reidy. 2007. *Statistics without maths for psychology*. Pearson education.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. 2021. On the effectiveness of adapter-based tuning for pretrained language model adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2208–2222, Online. Association for Computational Linguistics.

Leonid V Kantorovich. 1960. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422.

Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. 2008. Zero-data learning of new tasks. In *AAAI*, 2, page 3.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

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, Portland, Oregon, USA. Association for Computational Linguistics.

Abhinav Ramesh Kashyap, Devamanyu Hazarika, Min-Yen Kan, and Roger Zimmermann. 2021. Domain divergences: A survey and empirical analysis. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1830–1849, Online. Association for Computational Linguistics.

Alfréd Rényi et al. 1961. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, 547-561. Berkeley, California, USA.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with Bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382, Copenhagen, Denmark. Association for Computational Linguistics.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Edward H Simpson. 1949. Measurement of diversity. *nature*, 163(4148):688–688.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Brian MacWhinney, and Chris Dyer. 2016. Learning the curriculum with Bayesian optimization for task-specific word representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 130–139, Berlin, Germany. Association for Computational Linguistics.

Vincent Van Asch and Walter Daelemans. 2010. Using domain similarity for performance estimation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 31–36, Uppsala, Sweden. Association for Computational Linguistics.

Andrew K. C. Wong and Manlai You. 1985. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(5):599–609.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626.

A Implementation Details

Data Preparation. For each task, we sample a holdout validation set for early stopping and a test set, both of size 2,500 (10% of the maximum sample size), which remains fixed across both sets of experiments. After filtering by sample size, i.e. dropping domains with less than 30,000 samples, we had a total of 58 domains for comparison. Prior to sampling, the total number of domains are 43 and 22 for Amazon and Yelp datasets, respectively. We use the use the same five seeds for both data and model training.

Hyperparameters. We largely follow the recommended learning rate setting of 1e-4 (Pfeiffer et al., 2020; He et al., 2021) for adapter training. We set the max number of epochs to 50 and an early stopping patience of 5 non-decreasing epochs. We set the maximum input length to 256 and use a batch size of 32. For the adapter configuration, we make use of the PfeifferConfig (Pfeiffer et al., 2021) with default settings.

Computing Infrastructure. We run our experiments on 3× NVIDIA® TITAN™ RTX GPUs, with 130 Tensor TFLOPs of performance, 576 tensor cores, and 24 GB of GDDR6 memory. For the CPU, We use 4.5 cores of Intel® Xeon® Gold 5222 Processor (16.5MB Cache, 3.80 GHz) and 96GB of RAM, split across three docker containers.