Flexible Domain Adaptation for Automated Essay Scoring Using Correlated Linear Regression

Peter Phandi[1] **Kian Ming A. Chai**[2] **Hwee Tou Ng**[1]

[1] Department of Computer Science, National University of Singapore
{peter-p,nght}@comp.nus.edu.sg

[2] DSO National Laboratories
ckianmin@dso.org.sg

Abstract

Most of the current automated essay scoring (AES) systems are trained using manually graded essays from a specific prompt. These systems experience a drop in accuracy when used to grade an essay from a different prompt. Obtaining a large number of manually graded essays each time a new prompt is introduced is costly and not viable. We propose domain adaptation as a solution to adapt an AES system from an initial prompt to a new prompt. We also propose a novel domain adaptation technique that uses Bayesian linear ridge regression. We evaluate our domain adaptation technique on the publicly available Automated Student Assessment Prize (ASAP) dataset and show that our proposed technique is a competitive default domain adaptation algorithm for the AES task.

1 Introduction

Essay writing is a common task evaluated in schools and universities. In this task, students are typically given a prompt or essay topic to write about. Essay writing is included in high-stakes assessments, such as Test of English as a Foreign Language (TOEFL) and Graduate Record Examination (GRE). Manually grading all essays takes a lot of time and effort for the graders. This is what Automated Essay Scoring (AES) systems are trying to alleviate.

Automated Essay Scoring uses computer software to automatically evaluate an essay written in an educational setting by giving it a score. Work related to essay scoring can be traced back to 1966 when Ellis Page created a computer grading software called Project Essay Grade (PEG). Research on AES has continued through the years. The recent Automated Student Assessment Prize (ASAP) Competition[1] sponsored by the Hewlett Foundation in 2012 has renewed interest on this topic. The agreement between the scores assigned by state-of-the-art AES systems and the scores assigned by human raters has been shown to be relatively high. See Shermis and Burstein (2013) for a recent overview of AES.

AES is usually treated as a supervised machine learning problem, either as a classification, regression, or rank preference task. Using this approach, a training set in the form of human graded essays is needed. However, human graded essays are not readily available. This is perhaps why research in this area was mostly done by commercial organizations. After the ASAP competition, research interest in this area has been rekindled because of the released dataset.

Most of the recent AES related work is prompt-specific. That is, an AES system is trained using essays from a specific prompt and tested against essays from the same prompt. These AES systems will not work as well when tested against a different prompt. Furthermore, generating the training data each time a new prompt is introduced will be costly and time consuming.

In this paper, we propose domain adaptation as a solution to this problem. Instead of hiring people to grade new essays each time a new prompt is introduced, domain adaptation can be used to adapt the old prompt-specific system to suit the new prompt. This way, a smaller number of training essays from the new prompt is needed. In this paper, we propose a *novel* domain adaptation technique based on Bayesian linear ridge regression.

The rest of this paper is organized as follows. In Section 2, we give an overview of related work on AES and domain adaptation. Section 3 describes the AES task and the features used. Section 4 presents our novel domain adaptation algorithm.

[1] http://www.kaggle.com/c/asap-aes

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 431–439,
Lisbon, Portugal, 17-21 September 2015. ©2015 Association for Computational Linguistics.

Section 5 describes our data, experimental setup, and evaluation metric. Section 6 presents and discusses the results. We conclude in Section 7.

2 Related Work

We first introduce related work on automated essay scoring, followed by domain adaptation in the context of natural language processing.

2.1 Automated Essay Scoring

Since the first AES system, Project Essay Grade, was created in 1966, a number of commercial systems have been deployed. One such system, e-rater (Attali and Burstein, 2004), is even used as a replacement for the second human grader in the Test of English as a Foreign Language (TOEFL) and Graduate Record Examination (GRE). Other AES commercial systems also exist, such as IntelliMetric[2] and Intelligent Essay Assessor (Foltz et al., 1999).

AES is generally considered as a machine learning problem. Some work, such as PEG (Page, 1994) and e-rater, considers it as a regression problem. PEG uses a large number of features with regression to predict the human score. e-rater uses natural language processing (NLP) techniques to extract a smaller number of complex features, such as grammatical error and lexical complexity, and uses them with stepwise linear regression (Attali and Burstein, 2004). Others like (Larkey, 1998) take the classification approach. (Rudner and Liang, 2002) uses Bayesian models for classification and treats AES as a text classification problem. Intelligent Essay Assessor uses Latent Semantic Analysis (LSA) (Landauer et al., 1998) as a measure of semantic similarity between essays. Other recent work uses the preference ranking based approach (Yannakoudakis et al., 2011; Chen and He, 2013).

In this paper, we also treat AES as a regression problem, following PEG and e-rater. We use regression because the range of scores of the essays could be very large and a classification approach does not work well in this case. It also allows us to model essay scores as continuous values and scale them easily in the case of different score ranges between the source essay prompt and the target essay prompt.

The features used differ among the systems, ranging from simple features (e.g., word length, essay length, etc) to more complex features (e.g., grammatical errors). Some of these features are generic in the sense that they could apply to all kinds of prompts. Such features include the number of spelling errors, grammatical errors, lexical complexity, etc. Others are prompt-specific features such as bag of words features.

2.2 Domain Adaptation

The knowledge learned from a single domain might not be directly applicable to another domain. For example, a named entity recognition system trained on labeled news data might not perform as well on biomedical texts (Jiang and Zhai, 2007). We can solve this problem either by getting labeled data from the other domain, which might not be available, or by performing domain adaptation.

Domain adaptation is the task of adapting knowledge learned in a source domain to a target domain. Various approaches to this task have been proposed and used in the context of NLP. Some commonly used approaches include EasyAdapt (Daumé III, 2007), instance weighting (IW) (Jiang and Zhai, 2007), and structural correspondence learning (SCL) (Blitzer et al., 2006).

We can divide the approaches of domain adaptation into two categories based on the availability of labeled target data. The case where a small number of labeled target data is available is usually referred to as *supervised* domain adaptation (such as EasyAdapt and IW). The case where no labeled target domain data is available is usually referred to as *unsupervised* domain adaptation (such as SCL). In our work, we focus on *supervised* domain adaptation.

Daumé III (2007) described a domain adaptation scheme called EasyAdapt which makes use of feature augmentation. Suppose we have a feature vector $\boldsymbol{x}$ in the original feature space. This scheme will map this instance using the mapping functions $\Phi^{\mathrm{s}}(\boldsymbol{x})$ and $\Phi^{\mathrm{t}}(\boldsymbol{x})$ for the source and target domain respectively, where

$$\Phi^{\mathrm{s}}(x) = \langle \boldsymbol{x}, \boldsymbol{x}, \mathbf{0} \rangle$$
$$\Phi^{\mathrm{t}}(x) = \langle \boldsymbol{x}, \mathbf{0}, \boldsymbol{x} \rangle,$$

and $\mathbf{0}$ is a zero vector of length $|\boldsymbol{x}|$. This adaptation scheme is attractive because of its simplicity and ease of use as a pre-processing step, and also because it performs quite well despite its simplicity. It has been used in various NLP tasks such

[2] http://www.vantagelearning.com/products/intellimetric/

as word segmentation (Monroe et al., 2014), machine translation (Green et al., 2014), word sense disambiguation (Zhong et al., 2008), and short answer scoring (Heilman and Madnani, 2013). Our work is an extension of this scheme in the sense that our work is a generalization of EasyAdapt.

3 Automated Essay Scoring

This section describes the Automated Essay Scoring (AES) task and the features we use for the task.

3.1 Task Description

In AES, the input to the system is a student essay, and the output is the score assigned to the essay. The score assigned by the AES system will be compared against the human assigned score to measure their agreement. Common agreement measures used include Pearson's correlation, Spearman's correlation, and quadratic weighted Kappa (QWK). We use QWK in this paper, which is also the evaluation metric in the ASAP competition.

3.2 Features and Learning Algorithm

We model the AES task as a regression problem and use Bayesian linear ridge regression (BLRR) as our learning algorithm. We choose BLRR as our learning algorithm so as to use the correlated BLRR approach which will be explained in Section 4. We use an open source essay scoring system, EASE (Enhanced AI Scoring Engine)[3], to extract the features. EASE is created by one of the winners of the ASAP competition so the features they use have been proven to be robust. Table 1 gives the features used by EASE.

Useful n-grams are defined as n-grams that separate good scoring essays and bad scoring essays, determined using the Fisher test (Fisher, 1922). Good scoring essays are essays with a score greater than or equal to the average score, and the remainder are considered as bad scoring essays. The top 201 n-grams with the highest Fisher values are then chosen as the bag features. We perform the calculation of useful n-grams separately for source and target domain essays, and join them together using set union during the domain adaptation experiment. This is done to prevent the system from choosing only n-grams from the source domain as the useful n-grams, since the number of source domain essays is much larger than the target domain essays.

EASE uses NLTK (Bird et al., 2009) for POS tagging and stemming, aspell for spellchecking, and WordNet (Fellbaum, 1998) to get the synonyms. Correct POS tags are generated using a grammatically correct text (provided by EASE). The POS tag sequences not included in the correct POS tags are considered as bad POS. EASE uses scikit-learn (Pedregosa et al., 2011) for extracting unigram and bigram features. For linear regression, a constant feature of value one is appended for the bias.

4 Correlated Bayesian Linear Ridge Regression

First, consider the single-task setting. Let $\boldsymbol{x} \in \mathbb{R}^p$ be the feature vector of an essay. p represents the number of features in $\boldsymbol{x}$. The generative model for an observed real-valued score y is

$$\begin{aligned} &\alpha \sim \Gamma(\alpha_1, \alpha_2), && \lambda \sim \Gamma(\lambda_1, \lambda_2), \\ &\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \lambda^{-1} I), && f(\boldsymbol{x}) \overset{\text{def}}{=} \boldsymbol{x}^{\mathrm{T}} \boldsymbol{w}, \\ &y \sim \mathcal{N}(f(\boldsymbol{x}_i), \alpha^{-1}). \end{aligned}$$

Here, α and λ are Gamma distributed hyper-parameters of the model; $\boldsymbol{w} \in \mathbb{R}^p$ is the Normal distributed weight vector of the model; f is the latent function that returns the "true" score of an essay represented by $\boldsymbol{x}$ by linear combination; and y is the noisy observed score of $\boldsymbol{x}$.

Now, consider the two-task setting, where we indicate the source task and the target task by superscripts s and t. Given an essay with feature vector $\boldsymbol{x}$, we consider its observed scores y^{s} and y^{t} when evaluated in task s and task t separately. We have scale hyper-parameters α and λ sampled as before. In addition, we have the correlation ρ between the two tasks. The generative model relating the two tasks is

$$\begin{aligned} \rho &\sim p_\rho, \\ \boldsymbol{w}^{\mathrm{t}}, \boldsymbol{w}^{\mathrm{s}} &\sim \mathcal{N}(\boldsymbol{0}, \lambda^{-1} I), \\ f^{\mathrm{t}}(\boldsymbol{x}) &\overset{\text{def}}{=} \boldsymbol{x}^{\mathrm{T}} \boldsymbol{w}^{\mathrm{t}}, \\ f^{\mathrm{s}}(\boldsymbol{x}) &\overset{\text{def}}{=} \rho \boldsymbol{x}^{\mathrm{T}} \boldsymbol{w}^{\mathrm{t}} + (1-\rho^2)^{1/2} \boldsymbol{x}^{\mathrm{T}} \boldsymbol{w}^{\mathrm{s}}, \\ y^{\mathrm{t}} &\sim \mathcal{N}(f^{\mathrm{t}}(\boldsymbol{x}), \alpha^{-1}), \\ y^{\mathrm{s}} &\sim \mathcal{N}(f^{\mathrm{s}}(\boldsymbol{x}), \alpha^{-1}), \end{aligned}$$

where p_ρ is a chosen distribution over the correlation; and $\boldsymbol{w}^{\mathrm{t}}$ and $\boldsymbol{w}^{\mathrm{s}}$ are the weight vectors of the

[3] https://github.com/edx/ease

Feature Type	Feature Description
Length	Number of characters Number of words Number of commas Number of apostrophes Number of sentence ending punctuation symbols (".", "?", or "!") Average word length
Part of speech (POS)	Number of bad POS n-grams Number of bad POS n-grams divided by the total number of words in the essay
Prompt	Number of words in the essay that appears in the prompt Number of words in the essay that appears in the prompt divided by the total number of words in the essay Number of words in the essay which is a word or a synonym of a word that appears in the prompt Number of words in the essay which is a word or a synonym of a word that appears in the prompt divided by the total number of words in the essay
Bag of words	Count of useful unigrams and bigrams (unstemmed) Count of stemmed and spell corrected useful unigrams and bigrams

Table 1: Description of the features used by EASE.

target and the source tasks respectively, and they are identically distributed but independent. In this setting, it can be shown that the correlation between latent scoring functions for the target and the source tasks is ρ. That is,

$$\mathbb{E}(f^{\mathrm{t}}(\boldsymbol{x})f^{\mathrm{s}}(\boldsymbol{x}')) = \lambda^{-1}\rho\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}'. \qquad (1)$$

This, in fact, is a generalization of the EasyAdapt scheme, for which the correlation ρ is fixed at 0.5 [(Daumé III, 2007), see eq. 3]. Two other common values for ρ are 1 and 0; the former corresponds to a straightforward concatenation of the source and target data, while the latter is the shared-hyper-parameter setting which shares α and λ between the source and target domain. Through adjusting ρ, the model traverses smoothly between these three regimes of domain adaptation.

EasyAdapt is attractive because of its (frustratingly) ease of use via encoding the correlation within an expanded feature representation scheme. In the same way, the current setup can be achieved readily by the expanded feature representation

$$\begin{aligned} \Phi^{\mathrm{t}}(\boldsymbol{x}) &= \langle \boldsymbol{x}, \mathbf{0}_p \rangle, \\ \Phi^{\mathrm{s}}(\boldsymbol{x}) &= \left\langle \rho\boldsymbol{x}, (1-\rho^2)^{1/2}\boldsymbol{x} \right\rangle \end{aligned} \qquad (2)$$

in $\mathbb{R}^{2p}$ for the target and the source tasks. Associated with this expanded feature representation is the weight vector $\boldsymbol{w} \overset{\text{def}}{=} (\boldsymbol{w}^{\mathrm{t}}, \boldsymbol{w}^{\mathrm{s}})$ also in $\mathbb{R}^{2p}$. As we shall see in Section 4.1, such a representation eases the estimation of the parameters.

The above model is related to the multi-task Gaussian Process model that has been used for joint emotion analysis (Beck et al., 2014). There, the *intrinsic coregionalisation model* (ICM) has been used with *squared-exponential covariance function*. Here, we use the simpler *linear covariance function* (Rasmussen and Williams, 2006), and this leads to Bayesian linear ridge regression. There are two reasons for this choice. The first is that linear combination of carefully chosen features, especially lexical ones, usually gives good performance in NLP tasks. The second is in the preceding paragraph: an intuitive feature expansion representation of the domain adaptation process that allows ease of parameter estimation.

The above model is derived from the Cholesky decomposition

$$\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \rho & (1-\rho^2)^{1/2} \end{pmatrix} \begin{pmatrix} 1 & \rho \\ 0 & (1-\rho^2)^{1/2} \end{pmatrix}$$

of the desired correlation matrix that will eventually lead to equation (1). Other choices are possible, as long as equation (1) is satisfied. However, the current choice has the desired property that the $\boldsymbol{w}^{\mathrm{t}}$ portion of the combined weight vector is directly interpretable as the weights for the features in the target domain.

4.1 Maximum Likelihood Estimation

We estimate the parameters (α, λ, ρ) of the model using penalized maximum likelihood. For α and λ, the gamma distributions are used. For ρ, we impose a distribution with density $p_\rho(\rho) = 1 + a - 2a\rho$, $a \in [-1, 1]$. This distribution is supported only in $[0, 1]$; negative ρs are not supported because we think that negative transfer of information from source to target domain prompts in this essay scoring task is improbable. In our application, we slightly bias the correlations towards zero with $a = 1/10$ in order to ameliorate spurious correlations.

For the training data, let there be n^{t} examples in the target domain and n^{s} in the source domain. Let X^{t} (resp. X^{s}) be the n^{t}-by-p (resp. n^{s}-by-p) design matrix for the training data in the target (resp. source) domain. Let $\boldsymbol{y}^{\mathrm{t}}$ and $\boldsymbol{y}^{\mathrm{s}}$ be the corresponding observed essay scores. The expanded feature matrix due to equation (2) is

$$X \overset{\text{def}}{=} \begin{pmatrix} X^{\mathrm{t}} & 0 \\ \rho X^{\mathrm{s}} & (1-\rho^2)^{1/2} X^{\mathrm{s}} \end{pmatrix}.$$

Similarly, let $\boldsymbol{y}$ be the stacking of $\boldsymbol{y}^{\mathrm{t}}$ and $\boldsymbol{y}^{\mathrm{s}}$. Let $K \overset{\text{def}}{=} \lambda^{-1} X X^{\mathrm{T}} + \alpha^{-1} I$, which is also known as the Gramian for the observations. The log marginal likelihood of the training data is (Rasmussen and Williams, 2006)

$$L = -\frac{1}{2}\boldsymbol{y}^{\mathrm{T}} K^{-1} \boldsymbol{y} - \frac{1}{2}\log|K| - \frac{n^{\mathrm{t}} + n^{\mathrm{s}}}{2}\log 2\pi.$$

This is penalized to give L_p by adding

$$\begin{gathered}(\alpha_1 - 1)\log(\alpha) - \alpha_2\alpha + \alpha_1 \log\alpha_2 - \log\Gamma(\alpha_1) \\ +(\lambda_1 - 1)\log(\lambda) - \lambda_2\lambda + \lambda_1\log\lambda_2 - \log\Gamma(\lambda_1) \\ +\log(1 + a - 2a\rho).\end{gathered}$$

The estimation of these parameters is then done by optimising L_p. In our implementation, we use scikit-learn for estimating α and λ in an inner loop, and we use gradient descent for estimating ρ in the outer loop using

$$\frac{\partial L_p}{\partial \rho} = \frac{1}{2}\operatorname{tr}\left((\gamma\gamma^T - K^{-1}) \frac{\partial K}{\partial \rho} \right) - \frac{2a}{1 + a - 2a\rho},$$

where $\gamma \overset{\text{def}}{=} K^{-1} y$ and

$$\frac{\partial K}{\partial \rho} = \lambda^{-1} \begin{pmatrix} 0 & X^{\mathrm{t}}(X^{\mathrm{s}})^{\mathrm{T}} \\ X^{\mathrm{s}}(X^{\mathrm{t}})^{\mathrm{T}} & 0 \end{pmatrix}.$$

4.2 Prediction

We report the mean prediction as the score of an essay. This uses the mean weight vector $\bar{\boldsymbol{w}} = \lambda^{-1} X^{\mathrm{T}} K^{-1} \boldsymbol{y} \in \mathbb{R}^{2p}$, which may be partitioned into two vectors $\bar{\boldsymbol{w}}^{\mathrm{t}}$ and $\bar{\boldsymbol{w}}^{\mathrm{s}}$, each in $\mathbb{R}^p$. The prediction of a new essay represented by $\boldsymbol{x}_*$ in the target domain is then given by $\boldsymbol{x}_*^{\mathrm{T}} \bar{\boldsymbol{w}}^{\mathrm{t}}$.

5 Experiments

In this section, we will give a brief description of the dataset we use, describe our experimental setup, and explain the evaluation metric we use.

5.1 Data

We use the ASAP dataset[4] for our domain adaptation experiments. This dataset contains 8 prompts of different genres. The average length of the essays differs for each prompt, ranging from 150 to 650 words. The essays were written by students ranging in grade 7 to grade 10. All the essays were graded by at least 2 human graders. The genres include narrative, argumentative, or response. The prompts also have different score ranges, as shown in Table 2.

We pick four pairs of essay prompts to perform our experiments. In each experiment, one of the essay prompts from the pair will be the source domain and the other essay prompt will be the target domain. The essay set pairs we choose are 1 → 2, 3 → 4, 5 → 6, and 7 → 8, where the pair 1 → 2 denotes using prompt 1 as the source domain and prompt 2 as the target domain, for example. These pairs are chosen based on the similarities in their genres, score ranges, and median scores. The aim is to have similar source and target domains for effective domain adaptation.

5.2 Experimental Setup

We use 5-fold cross validation on the ASAP training data for evaluation. This is because the official test data of the competition is not released to the public. We divide the target domain data randomly into 5 folds. One fold is used as the test data, while the remaining four folds are collected together and then sub-sampled to obtain the target-domain training data. The sizes of the sub-sampled target-domain training data are 10, 25, 50 and 100, with the larger sets containing the smaller sets. All essays from the source domain are used.

[4] https://www.kaggle.com/c/asap-aes/data

				Score	
Set	# Essays	Genre	Avg len	Range	Median
1	1,783	ARG	350	2–12	8
2	1,800	ARG	350	1–6	3
3	1,726	RES	150	0–3	1
4	1,772	RES	150	0–3	1
5	1,805	RES	150	0–4	2
6	1,800	RES	150	0–4	2
7	1,569	NAR	250	0–30	16
8	723	NAR	650	0–60	36

Table 2: Selected details of the ASAP data. For the genre column, ARG denotes *argumentative* essays, RES denotes *response* essays, and NAR denotes *narrative* essays.

Our evaluation considers the following four ways in which we train the AES model:

SourceOnly Using essays from the source domain only;

TargetOnly Using 10, 25, 50, and 100 sampled essays from the target domain only;

SharedHyper Using correlated Bayesian linear ridge regression (BLRR) with ρ fixed to 0 on source domain essays and sampled essays from the target domain.

EasyAdapt As SharedHyper, but with $\rho = 0.5$;

Concat As SharedHyper, but with ρ fixed to 1.0;

ML-ρ Using correlated BLRR with ρ maximizing the likelihood of the data.

Since the source and target domain may have different score ranges, we scale the scores linearly to range from -1 to 1. When predicting on the test essays, the predicted scores of our system will be linearly scaled back to the target domain score range and rounded to the nearest integer.

We build upon scikit-learn's implementation of BLRR for our learning algorithm. To ameliorate the effects of different scales of features, we normalize the features: length, POS, and prompt features are linearly scaled to range from 0 to 1 according to the training data; and the feature values for bag-of-words features are $\log(1 + \text{count})$ instead of the actual counts.

We use scikit-learn version 0.15.2, NLTK version 2.0b7, and aspell version 0.60.6.1 in this experiment. The BLRR code (`bayes.py`) in scikit-learn is modified to obtain valid likelihoods for use in the outer loop for estimating ρ. We use scikit-learn's default value for the parameters α_1, α_2, λ_1, and λ_2 which is 10^{-6}.

	QWK scores		
Set #	BLRR	SVM	Human
1	0.761	0.781	0.721
2	0.606	0.621	0.814
3	0.621	0.630	0.769
4	0.742	0.749	0.851
5	0.784	0.782	0.753
6	0.775	0.771	0.776
7	0.730	0.727	0.721
8	0.617	0.534	0.629

Table 3: In-domain experimental results.

5.3 Evaluation Metric

Quadratic weighted Kappa (QWK) is used to measure the agreement between the human rater and the system. We choose to use this evaluation metric since it is the official evaluation metric of the ASAP competition. Other work such as (Chen and He, 2013) that uses the ASAP dataset also uses this evaluation metric. QWK is calculated using

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}},$$

where matrices O, $(w_{i,j})$, and E are the matrices of observed scores, weights, and expected scores respectively. Matrix $O_{i,j}$ corresponds to the number of essays that receive a score i by the first rater and a score j by the second rater. The weight entries are $w_{i,j} = (i-j)^2/(N-1)^2$, where N is the number of possible ratings. Matrix E is calculated by taking the outer product between the score vectors of the two raters, which are then normalized to have the same sum as O.

6 Results and Discussion

In-domain results for comparison First, we determine indicative upper bounds on the QWK scores using Bayesian linear ridge regression (BLRR). To this end, we perform 5-fold cross validation by training and testing within each domain. This is also done with linear support vector machine (SVM) regression to confirm that BLRR is a competitive method for this task. In addition, since the ASAP data has at least 2 human annotators for each essay, we also calculate the human agreement score. The results are shown in Table 3. We see that the BLRR scores are close to the the human agreement scores for prompt 1 and

	QWK Scores			
Method	$n^{\mathrm{t}}=10$	25	50	100
$1 \rightarrow 2$				
SourceOnly	0.434			
TargetOnly	0.069	0.169	0.279	0.395
SharedHyper	0.158	0.218	0.332	0.390
EasyAdapt	0.425	0.422	0.442	0.467
Concat	**0.484**	**0.507**	**0.529**	**0.545**
ML-ρ	0.463	0.457	0.492	0.510
$3 \rightarrow 4$				
SourceOnly	0.522			
TargetOnly	0.117	0.398	0.545	0.626
SharedHyper	0.113	0.350	0.487	0.575
EasyAdapt	0.461	0.541	0.589	0.628
Concat	**0.594**	**0.611**	0.617	0.638
ML-ρ	0.593	0.609	**0.618**	**0.646**
$5 \rightarrow 6$				
SourceOnly	0.187			
TargetOnly	0.416	0.506	0.554	0.608
SharedHyper	0.380	0.500	0.544	0.600
EasyAdapt	0.553	0.621	0.652	0.698
Concat	**0.649**	**0.689**	**0.708**	**0.722**
ML-ρ	0.539	0.662	0.680	0.713
$7 \rightarrow 8$				
SourceOnly	0.171			
TargetOnly	0.290	0.381	0.426	0.477
SharedHyper	0.302	0.383	0.444	0.484
EasyAdapt	**0.594**	**0.616**	0.605	0.610
Concat	0.332	0.362	0.396	0.463
ML-ρ	0.586	0.607	**0.613**	**0.621**

Table 4: QWK scores of the six methods on four domain adaptation experiments, ranging from using 10 target-domain essays (second column) to 100 target-domain essays (fifth column). The scores are the averages over 5 folds. Setting $a \rightarrow b$ means the AES system is trained on essay set a and tested on essay set b. For each set of six results comparing the methods, the best score is boldfaced and the second-best score is underlined.

prompts 5 to 8, but fall short by 10% to 20% for prompts 2 to 4. We also see that BLRR is comparable to linear SVM regression, giving almost the same performance for prompts 4 to 7; slightly poorer performance for prompts 1 to 3; and much better performance for prompt 8. The subsequent discussion in this section will refer to the BLRR scores in Table 3 for in-domain scores.

Importance of domain adaptation The results of the domain adaptation experiments are tabulated in Table 4, where the best scores are boldfaced and the second-best scores are underlined. As expected, for pairs 1 → 2, 3 → 4, and 5 → 6, all the scores are below their corresponding upper bounds from the in-domain setting in Table 3. However, for pair 7 → 8, the QWK score for domain adaptation with 100 target essays outperforms that of the in-domain, albeit only by 0.4%. This can be explained by the small number of essays in prompt 8 that can be used in both the in-domain and domain adaptation settings, and that domain adaptation additionally involves prompt 7 which has more than twice the number of essays; see column two in Table 2. Hence, domain adaptation is effective in the context of small number of target essays with large number of source essays. This can also be seen in Table 4 where we have simulated small number of target essays with sizes 10, 25, 50, and 100. When we compare the scores of TargetOnly against the best scores and second-best scores, we find that domain adaptation is effective and important in improving the QWK scores.

By the above argument alone, one might have thought that an overwhelming large number of source domain essays was sufficient for the target domain. However, this is not true. When we compare the scores of SourceOnly against the best scores and second-best scores, we find that domain adaptation again improves the QWK scores. In fact, with just 10 additional target domain essays, effective domain adaptation can improve over SourceOnly for all target domains 2, 4, 6, and 8 respectively.

This is the first time where the effects of domain adaptation are shown in the AES task. In addition, the large improvement with a small number of additional target domain essays in 5 → 6 and 7 → 8 suggests the high domain-dependence nature of the task: *learning on one essay prompt and testing on another should be strongly discouraged.*

Contributions by target-domain essays It is instructive to understand why domain adaptation is important for AES. To this end, we estimate the contribution of bag-of-words features to the overall prediction by computing the ratio

$$\frac{\sum_{i \text{ over bag-of-words features}} w_i^2}{\sum_{i \text{ over all features}} w_i^2}$$

using weights learned in the in-domain setting; see Table 1 for the complete list of features. For domains 2, 4, 6, and 8, which are the target domains in the domain adaptation experiments, these ratios are 0.37, 0.73, 0.69, and 0.93. The ratios for the other four domains are similarly high. This shows that bag-of-words features play a significant role in the prediction of the essay scores. We examine the number of bag-of-words features that 100 additional target domain essays would add to SourceOnly; that is, we compare the bag-of-words features for SourceOnly with those of SharedHyper, EasyAdapt, Concat, and ML-ρ for $n^{\mathrm{t}} = 100$. The numbers of these additional features, averaged over the five folds, are 269, 351, 377, and 291 for target domains 2, 4, 6, and 8 respectively. In terms of percentages, these are 67%, 87%, 94%, and 72% more features over SourceOnly. Such a large number of additional bag-of-words features contributed by target-domain essays, together with the fact that these features are given high weights, means that target-domain essays are important.

Comparing domain adaptation methods We now compare the four domain adaptation methods: SharedHyper, EasyAdapt, Concat, and ML-ρ. We recall that the first three are constrained cases of the last by fixing ρ to 0, 0.5, and 1 respectively. First, we see that SharedHyper is a rather poor domain adaptation method for AES, because it gives the lowest QWK score, except for the case of using 25, 50, and 100 target essays in adapting from prompt 7 to prompt 8, where it is better than Concat. In fact, its scores are generally close to the TargetOnly scores. This is unsurprising, since in SharedHyper the weights are effectively not shared between the target and source training examples: only the hyper-parameters α and λ are shared. This is a weak form of information sharing between the target and source domains. Hence, we expect this to perform suboptimally when the target and source domains bear more than spurious relationship, which is indeed the case here because we have chosen the source and target domain pairs based on their similarities, as described in Section 5.1.

We now focus on EasyAdapt, Concat, and ML-ρ, which are the better domain adaptation methods from our results. We see that ML-ρ either gives the best or second-best scores, except for the one case of $5 \rightarrow 6$ with 10 target essays. In comparison, although Concat performs consistently well for $1 \rightarrow 2$, $3 \rightarrow 4$, and $5 \rightarrow 6$, its QWK scores for $7 \rightarrow 8$ are quite poor and even lower than those of TargetOnly for 25 or more target essays. In contrast to Concat, EasyAdapt performs well for $7 \rightarrow 8$ but not so well for the other three domain pairs.

Let us examine the reason for contrasting results between EasyAdapt and Concat to appreciate the flexibility afforded by ML-ρ. The ρ estimated by ML-ρ for the pairs $1 \rightarrow 2$, $3 \rightarrow 4$, $5 \rightarrow 6$, and $7 \rightarrow 8$ with 100 target essays are 0.81, 0.97, 0.76, and 0.63 averaged over five folds. The lower estimated correlation ρ for $7 \rightarrow 8$ means that prompt 7 and prompt 8 are not as similar as the other pairs are. In such a case as this, Concat, which in effect considers the target domain to be exactly the same as the source domain, can perform very poorly. For the other three pairs which are more similar, the correlation of 0.5 assumed by EasyAdapt is not strong enough to fully exploit the similarities between the domains. Unlike Concat and EasyAdapt, ML-ρ has the flexibility to allow it to traverse effectively between the different degrees of domain similarity or relatedness based on the source domain and target domain training data. In view of this, we consider *ML-ρ to be a competitive default domain adaptation algorithm for the AES task.*

In retrospect of our present results, it can be obvious why prompts 7 and 8 are not as similar as we would have hoped for more effective domain adaptation. Both prompts ask for narrative essays, and these by nature are very prompt-specific and require words and phrases relating directly to the prompts. In fact, referring to a previous discussion on the *contributions by target-domain essays*, we see that weights for the bag-of-words features for prompt 8 contribute a high of 93% of the total. When we examine the bag-of-words features, we see that prompt 7 (which is to write about patience) contributes only 19% to the bag-of-words features of prompt 8 (which is to write about laughter) in the in-domain experiment. This means that 81% of the bag-of-words features, which are important to narrative essays, must be contributed by the target-domain essays relating to prompt 8. Future work on domain adaptation for AES can explore chosing the prior p_ρ on ρ to better reflect the nature of the essays involved.

7 Conclusion

In this work, we investigate the effectiveness of using domain adaptation when we only have a small number of target domain essays. We have shown that domain adaptation can achieve better results compared to using just the small number of target domain data or just using a large amount of data from a different domain. As such, our research will help reduce the amount of annotation work needed to be done by human graders to introduce a new prompt.

Acknowledgments

This research is supported by Singapore Ministry of Education Academic Research Fund Tier 2 grant MOE2013-T2-1-150.

References

Yigal Attali and Jill Burstein. 2004. Automated essay scoring with e-rater® v. 2.0. Technical report, Educational Testing Service.

Daniel Beck, Trevor Cohn, and Lucia Specia. 2014. Joint emotion analysis via multi-task Gaussian processes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*.

Hongbo Chen and Ben He. 2013. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. Bradford.

Ronald A Fisher. 1922. On the interpretation of χ^2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*.

Peter W Foltz, Darrell Laham, and Thomas K Landauer. 1999. The intelligent essay assessor: Applications to educational technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*.

Spence Green, Daniel Cer, and Christopher D. Manning. 2014. An empirical comparison of features and tuning for phrase-based machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*.

Michael Heilman and Nitin Madnani. 2013. Ets: domain adaptation and stacking for short answer scoring. In *Proceedings of the Seventh International Workshop on Semantic Evaluation*.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes*.

Leah S Larkey. 1998. Automatic essay grading using text categorization techniques. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Will Monroe, Spence Green, and Christopher D Manning. 2014. Word segmentation of informal Arabic with domain adaptation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Ellis Batten Page. 1994. Computer grading of student prose, using modern concepts and software. *The Journal of Experimental Education*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*.

Carl Edward Rasmussen and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. MIT Press.

Lawrence M Rudner and Tahung Liang. 2002. Automated essay scoring using Bayes' theorem. *The Journal of Technology, Learning and Assessment*.

Mark D. Shermis and Jill Burstein, editors. 2013. *Handbook of Automated Essay Evaluation: Current Applications and New Directions*. Routledge.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

Zhi Zhong, Hwee Tou Ng, and Yee Seng Chan. 2008. Word sense disambiguation using OntoNotes: An empirical study. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*.