Domain Adaptation for Automated Essay Scoring

Peter Phandi

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

2016

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Peter Phandi

15 December 2016

Acknowledgments

First of all, I would like to thank God. His grace and blessings have enabled me to complete the work in this thesis.

I would like to express my gratitude to my supervisor, Professor Ng Hwee Tou, for his guidance and support. Without him, this thesis would not have been possible.

I would also like to thank Kian Ming Adam Chai for his collaboration. His knowledge and insights of machine learning have cultivated my interest in this area.

I would like to thank my lab-mates in the NUS Natural Language Processing group: Benjamin Yap, Christopher Bryant, Christian Hadiwinoto, Kaveh Taghipour, Nhu Thao Nguyen, Raymond Hendy Susanto, Shamil C.M., Tam Hoang and Wang Hongmin; for the meaningful discussion and for all the work that we have done together. Thanks to them, the NLP lab has been a comfortable place to work in.

Finally, I would like to thank my parents for their support. They have always been there for me whenever I need them.

Contents

Summary

Automated Essay Scoring (AES) is an important task in Natural Language Processing. The research done by various commercial organizations has identified the features that correlate well with human scoring. They have built strong AES systems that achieve high agreement with human scoring based on these features. One of these commercial organizations, ETS, even uses their own AES system (e-rater) as a second rater for their high-stakes exams, GRE and TOEFL. However, most of these AES systems use prompt-specific features. This means that each time a new prompt is introduced, a large number of essays need to be annotated as training data. This thesis gives an overview of the AES task and shows that domain adaptation can help an AES system to achieve high performance with a small number of annotated essays.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Essay Writing

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. The students will then receive feedback from the grader in the form of a score. The score is typically given following some sort of marking criteria called rubrics. Some example prompts can be seen in Appendix A.

The essay writing task is included in high-stakes assessments, such as Test of English as a Foreign Language (TOEFL) and Graduate Record Examination (GRE). The results of these high-stakes assessments are usually used as an entrance requirement for universities or colleges.

In order to improve their writing skills, students will need to write more essays. However, manually grading students' essays takes a lot of time and effort for the graders. This is what leads to the development of Automated Essay Scoring (AES) systems. By automating the grading process, students will be able to practise their writing more.

1.2 Automated Essay Scoring

Automated Essay Scoring is the task of assigning a score to a student essay. In this task, the input will be a student essay and the output will be the predicted score of that essay. Most of the work in this task was done for English essays, but the techniques have also been applied to Japanese, Hebrew, and Malay essays (Shermis and Burstein, 2013).

AES benefits from other Natural Language Processing (NLP) tasks, such as grammatical error correction, discourse parsing, and sentiment analysis. Grammatical errors are useful to measure the fluency of an essay. Discourse parsing can be used to determine a measure of discourse coherence which is an important aspect of quality writing. Sentiment analysis is useful in evaluating argumentative essays. It can help to identify the writer's opinions which are relevant in argument construction in essay writing.

Some of the advantages of AES are its speed and reliability. Using the computer, an AES system can analyze significantly more essays than humans can in the same amount of time. It is reliable because it will return the same score for the same essay all the time, which may not be true for human scoring. There are some disadvantages of AES which will be explained in a later section.

The rubric used by human evaluators can be classified into two categories: holistic rubrics and trait rubrics. A holistic rubric measures the overall quality of a writer's performance, while a trait rubric measures multiple traits of writing. Some AES systems have the ability to grade using both holistic and trait rubrics.

Holistic rubics have the advantages of efficiency and reliability. It has been shown that humans achieve a higher agreement when grading using a holistic approach (Page, Poggio, and Keith, 1997). Using a holistic approach, humans only need to judge the overall performance of the writer. This increases the efficiency of scoring.

The main disadvantage of the holistic approach is that the writer cannot easily determine which writing traits he is lacking. This is what a trait rubric tries to address.

1.2.1 History

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) (Shermis and Burstein, 2013). The project faced an immense difficulty at the time. The main input medium was IBM punch cards and there was no text processing software during the 1960s. Also, computers were not as widespread as now, making this invention too costly for any average student to use. These reasons caused the development of AES to stall for a decade. The development continued during the subsequent decades due to advances in computer technology, particularly the availability of word processing applications, the advances of NLP, and the advent of the Internet.

Word processing software makes it easier for students to write their essays using computer, instead of using pen and paper. Advances in NLP techniques mean that AES systems can now use more sophisticated features and score essays more accurately. Finally, the Internet allows students to upload their essays and check their scores online easily. This was done by PEG in 1998. (Shermis et al., 2001)

Since the very beginning, the development of AES systems was mostly done by commercial or non-profit organizations. The essays used by these organizations come from their customers. These organizations need to protect the privacy of their customers. Hence, there is a lack of publicly available AES datasets.

A number of commercial AES systems have been deployed, including the first AES system, Project Essay Grade. The AES system developed by ETS, e-rater (At-

tali and Burstein, 2004), is 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[1] and Intelligent Essay Assessor (Foltz, Laham, and Landauer, 1999). In 2001, the first publicly available AES system, Bayesian Essay Test Scoring sYstem (BETSY), was released. More recently, the open-source AES systems LightSIDE and EASE were also publicly released.

1.2.2 Challenges

One of the challenges of automated essay scoring is the connotation raised by "writing to machines" instead of humans. There are some commonly mentioned concerns regarding AES. For example,"Can the system be gamed?", "Does using AES systems encourage the students to focus only on some aspects of writing, which are the ones detected by the AES system?", and "Does writing to the machines subvert the intention of the writing itself, which is to convey your thoughts to other humans?" These concerns focus on whether the writer truly is writing to convey his thought, or whether he is just writing to achieve a good score by following certain guidelines. However, this same criticism also applies to some types of academic writing, such as expository and persuasive writing (Connors, 1981).

AES systems also need a significant amount of data for training. This may not be a problem for commercial organizations, since they have a lot of essays from their customers. However, for the research community, student essays are not so readily available.

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

1.2.3 Automated Student Assessment Prize (ASAP)

In 2012, the Hewlett Foundation sponsored the Automated Student Assessment Prize (ASAP)[2], which fuels research interest in AES. ASAP is a competition in Kaggle to create the best AES system. The competition is significant in that it brings AES to the public's attention and releases a new dataset for AES. This competition also serves as an evaluation of the established AES vendors. They have shown that the best vendor systems can achieve scores that are relatively close to the scores assigned by humans. Moreover, one of the winners of the ASAP competition, EASE, releases their code as an open source package.

The ASAP organizers released a dataset that contains student essays written for 8 different prompts. The prompts are included in Appendix A. The essays were written by students ranging from grade 7 to grade 10. The prompts have different genre, score range, and grading criteria. The variability makes this a good dataset for evaluating AES systems. Table 1.1 gives the details of the ASAP dataset.

The rest of this thesis is organized as follows. In Chapter 2, we give an overview of current approaches to AES. Chapter 3 describes the prompt specificity problem in AES that we are trying to address. Chapter 4 presents our novel domain adaptation algorithm, which is the main contribution of this thesis. Chapter 5 describes our data, experimental setup, and evaluation metric. Chapter 6 presents and discusses the results. Finally, we conclude in Chapter 7 and suggests future directions for AES.

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

				Score	
Prompt	# Essays	Genre	Avg length	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 1.1: Details of the ASAP dataset. For the genre column, ARG denotes *argumentative* essays, RES denotes *response* essays, and NAR denotes *narrative* essays.

Chapter 2

Approaches to AES

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

2.1 Commercial AES Software

In this section, we will discuss the approaches of some commercial AES systems.

2.1.1 Project Essay Grade

Project Essay Grade (Page, 1994) models the AES task as a regression problem. As the pioneer of AES technology, it only uses shallow NLP features such as parts of speech (POS). The work uses 30–40 predictors as features to a linear regression algorithm.

The work does not specify exactly which features are used, but it notes that essay length is an important feature in their system. They use the fourth root of essay length instead of the actual value because the relationship of essay length and score is not linear. Essay length will be considered only up to a certain threshold and then other aspects of writing will be evaluated more.

The work also compares the correlation score against human agreement and shows that PEG can achieve comparable or even better agreement compared to the human agreement score. This shows the feasibility of using computers to grade essays.

2.1.2 Intelligent Essay Assessor

In 1998, Pearson Knowledge Technologies created an AES system, Intelligent Essay Assessor (IEA) (Landauer, Laham, and Foltz, 2003), which uses Latent Semantic Analysis (LSA). LSA represents documents as a 2 dimensional term-document matrix containing the frequency of each term in each document. The matrix is then decomposed using Singular Value Decomposition (SVD) to obtain a low-rank approximation of the matrix and a reduced vector representation of the documents. The documents can then be compared based on the cosine similarity of their vector representation.

IEA uses LSA to compare an essay against a set of training examples and find similar essays in the training examples. The essay is assigned a content score based on the average score of its similar essays. LSA can also be used to compare

individual sentences against each other as a measure of essay coherence.

In addition to LSA features, IEA also uses other features which correspond to different aspects in writing. Below is the list of the feature categories used by IEA:

- Content features
- Mechanics features
- Grammar features
- Style, organization, and development features
- Lexical sophistication features

Using these features, in addition to the holistic essay score, IEA is able to score different traits of writing, such as content, development, etc. This will be useful to the students, because it provides a direction on the aspects of writing that they need to improve upon.

2.1.3 e-rater

e-rater was created by Educational Testing Service (ETS), which conducts the high-stakes assessments TOEFL and GRE. It uses NLP techniques to extract features that are related to the human evaluation rubric and empirically correlate highly with the human-assigned essay score. e-rater is used by Criterion, the ETS Online Writing Evaluation Service.

The older version of e-rater (v1.3) has a large pool of about 50 features. It filters them using stepwise linear regression. Stepwise linear regression is a linear regression technique that can determine the choice of predictive features automatically. It does so by a backward elimination procedure. Starting with the full feature set, the algorithm tries removing each feature one at a time and finally eliminates

one feature that gives the highest improvement to the model when it is removed. This process is repeated until no further improvement could be made to the model.

After further analysis, they discovered that a lot of the older features implicitly measures the essay length. After filtering out those features, they came up with a smaller set of features that correlates well with human-assigned score for e-rater v2.0. These features can be grouped into several feature classes:

- Errors in grammar, usage, mechanics, and style

  The rates of errors in grammar, usage, machanics, and style.

- Organization and development

  They separate the sentences in an essay into several categories: background, thesis, main ideas, supporting ideas, and conclusion. Features are then derived from these categories (e.g., presence of thesis, main ideas, supporting ideas, and conclusion).

- Lexical complexity

  This is derived using the ratio of unique words in the essay. They also use the Breland's standardized frequency index (Breland et al., 1994) to measure the level of the vocabulary used.

- Prompt-specific vocabulary usage

  This is done using Content Vector Analysis (CVA). It compares the vocabulary usage of an essay with a manually graded model example.

- Essay length

e-rater models the AES task as a regression problem and tackles it using linear regression.

2.2 Open-source AES Software

We will discuss the open-source AES Systems EASE and LightSIDE in this section.

2.2.1 EASE

EASE is a system developed by one of the winners of the ASAP competition. It is written in Python and uses the scikit-learn library (Pedregosa et al., 2011). It uses the gradient boosting algorithm and models AES as a regression problem. Table 2.1 lists the features used by EASE.

Gradient boosting is a machine learning algorithm that uses an ensemble of weaker models to create a stronger model. The scikit-learn implementation used by EASE uses decision trees as the weaker models and mean squared error as the loss function. Gradient boosting starts with a basic weak model and improves upon it iteratively using gradient descent. In the EASE setting, the model F_0 will start with a single decision tree h_0. On each step m of gradient boosting, we want to add another decision tree h_{m+1} so that the prediction of the new model F_{m+1} is better than the previous model F_m, where $F_{m+1}(x) = F_m(x) + h_{m+1}(x)$. Ideally, the better model will predict the observed class perfectly ($F_{m+1}(x) = y$). So, we need to fit h_{m+1} to the *residual* $y - F_m(x)$, so that $h_{m+1} = y - F_m(x)$. Notice that $y - F_m(x)$ is the negative gradient of the mean squared error loss function $\frac{1}{2}(y - F_m(x))^2$. Finally, we add a step length parameter γ_{m+1} such that $F_{m+1}(x) = F_m(x) + \gamma_{m+1} h_{m+1}(x)$. The step length is chosen using line search to minimize the loss function.

Useful n-grams are defined as n-grams that separate good scoring essays from bad scoring essays, determined using the Fisher test (Fisher, 1922). Good scoring essays are essays with a score greater than or equal to the median score, and the remaining essays are considered as bad scoring essays. The top 201 n-grams with the highest Fisher values are then chosen as the bag-of-words features.

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 appear in the prompt • Number of words in the essay that appear in the prompt divided by the total number of words in the essay • Number of words w in the essay such that w is a word or a synonym of a word that appears in the prompt • Number of words w in the essay such that w 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 2.1: Description of the features used by EASE.

EASE uses NLTK (Bird, Klein, and Loper, 2009) for part-of-speech (POS) tagging and stemming, Aspell for spell-checking, and WordNet (Fellbaum, 1998) to get the synonyms. Correct POS tag sequences are generated using a grammatically correct text (provided by EASE). The text consists of various novels taken from the public domain ebook provider *gutenberg.org*. The POS tag sequences not included in the correct POS tag sequences are considered as bad POS tag sequences. EASE uses scikit-learn for extracting unigram and bigram features.

2.2.2 LightSIDE

LightSIDE is a machine learning system for automatic text evaluation. Instead of creating a specific set of features focused on the AES task, LightSIDE provides general features for general text evaluation. It provides basic features such as text length and bag-of-words features. It also provides bigrams, POS tags, stop-word removal, and stemming. LightSIDE allows the user to easily add new features using plugin and with minimal programming. LightSIDE can handle both regression and classification models. It gives the user the liberty of choosing which machine learning algorithm to use, primarily using the implementation from Weka (Hall et al., 2009).

LightSIDE is one of the vendors invited to the ASAP competition. In this competition, LightSIDE uses only unigram features and POS bigram features. It uses only the top 500 predictive features which are chosen by using the chi square statistical test. It models AES as a classification task and uses naïve Bayes as the classifier. Other learning algorithms were also tried, but they gave worse performance.

2.3 Other AES Approaches

2.3.1 Classification Approaches

(Larkey, 1998) compares 3 different algorithms: Bayesian Independence (BI) Classifier, K-nearest Neighbor (KNN) Classifier, and Stepwise Linear Regression. They train several binary BI classifiers, dividing the score range into several classes. For essays with 4 point scores, they train a binary classifier to distinguish 1's from 2's, 3's, and 4's, another one to distinguish "1's and 2's" from "3's and 4's", and finally one to distinguish "1's, 2's, and 3's" from 4's. For the KNN approach, they compare essays using the Inquery Retrieval System (Callan, Croft, and Harding, 1992). When grading an essay, they choose the k-most similar essays in the training set and assign their average score to the essay to be graded. The stepwise linear regression approach uses several text-complexity features and combines the BI and KNN approach by using their output as features. They found that the BI approach achieves the best performance out of the 3 approaches and also show that it has a similar score with human graders.

(Rudner and Liang, 2002) uses Bayesian models for classification and treats AES as a text classification problem. They compare two Bayesian models typically used in text classification: multivariate Bernoulli model and multinomial model. The Bernoulli model checks the presence of a term in an essay, while multinomial model counts how many times a term appears in an essay. They use unigram, bigram, and argument (non-adjacent bigram) features. They perform feature selection based on information gain and feature prevalence (the number of occurrences per 1,000 essays). They perform the experiment for both the stemmed and unstemmed version of the features. Their experiment shows that the Bernoulli model tends to outperform the multinomial model, unstemmed features tend to outperform stemmed and stopword-removed features, and some feature selection using

feature prevalence improves the model.

2.3.2 Ranking Approaches

(Yannakoudakis, Briscoe, and Medlock, 2011) treats AES as a pairwise ranking preference problem. They argue that learning a ranking directly is a more generic approach to the AES problem, compared to fitting a classifier score. They choose Support Vector Machine (SVM) as their learning algorithm and use the open source SVM^{light} software (Joachims, 1999). The features they use in this work include: word n-grams, POS n-grams, syntactic features, script length, and error rate features. They train and compare 2 models, SVM rank and SVM regression, using the same set of features. They evaluate using Pearson's and Spearman's correlation and show that the SVM rank model is superior. They released a new dataset for AES, extracted from the essays written for the First Certificate in English (FCE) examination. Each essay in the corpus has about 200–400 words and is annotated with marks in the range of 1 to 40.

(Chen and He, 2013) uses a listwise ranking approach to AES, instead of the pairwise ranking approach used by (Yannakoudakis, Briscoe, and Medlock, 2011). The listwise ranking approach processes a list of essays each time and tries to achieve the best agreement between the predicted ranking and the actual scores. The work aims to create an AES model that maximizes human and machine agreement. Since the listwise ranking approach takes the whole list of essay scores as the training input, they can modify the loss function to include the inter-rater agreement. They use the LambdaMART (Wu et al., 2008) algorithm with random forest bagging which has been widely used in Information Retrieval. They modify the loss function of LambdaMART by multiplying it with the quadratic weighted kappa (QWK), which is the evaluation metric used in the ASAP competition. They use 4 groups of features: lexical features, syntactic features, grammar and fluency

features, and content and prompt-specific features. They evaluate their algorithm on the ASAP dataset and compare it with SVM regression, classification, and ranking algorithms. They show that their algorithm outperforms those baselines and achieves a high QWK score. They also perform some experiment on a generic model which will be elaborated more on Chapter 3.

2.3.3 Trait-based Approaches

Most of the aforementioned AES systems focus on producing a holistic score for each essay. However, the main use of an AES system is to help users practise their essay writing. A holistic score alone is not sufficient for a user to know which trait they should improve their writing on. There has been some work on essay scoring that focuses only on one trait of essay quality at a time, such as coherence (Miltsakaki and Kukich, 2004; Burstein, Tetreault, and Andreyev, 2010; Yannakoudakis and Briscoe, 2012), organization (Persing, Davis, and Ng, 2010), prompt adherence (Persing and Ng, 2014; Higgins et al., 2004), thesis clarity (Persing and Ng, 2013), and argument strength (Persing and Ng, 2015). Most of the work in this area focuses on annotation and feature engineering. Isaac Persing's work annotates the International Corpus of Learner's English (ICLE) (Granger et al., 2009) on different traits of essay quality and also does some feature engineering on them. The annotation is publicly available at his website[3]. His work is important since it provides the dataset and an evaluation model for the trait-based approaches.

[3]http://www.hlt.utdallas.edu/~persingq/ICLE/

Chapter 3

Prompt Specificity

As we can see from the previous chapter, AES systems contain prompt-specific features. Each time a new prompt is introduced, scored essays specific to that prompt need to be collected to serve as training data. This is done by getting some annotators to score essays for the new prompt. ETS has noted that they need at least 500 essays to build a new model (Attali and Burstein, 2004). There have been some efforts to fix this problem, namely by creating a generic model (Attali, Bridgeman, and Trapani, 2010; Zesch, Wojatzki, and Scholten-Akoun, 2015) or by using domain adaptation as we have done recently (Phandi, Chai, and Ng, 2015).

3.1 Generic Model

The approach of using a generic model aims to create a model that will not have a significant drop in performance when used for another prompt. This is typically done by using only features that are not prompt-specific. We will discuss the approaches by (Attali, Bridgeman, and Trapani, 2010), (Chen and He, 2013), and (Zesch, Wojatzki, and Scholten-Akoun, 2015).

3.1.1 The ETS Generic Model

(Attali, Bridgeman, and Trapani, 2010) describes ETS' effort to create a generic model for their e-rater. ETS achieves this by removing their prompt-specific vocabulary features that use content vector analysis. They evaluated 3 models: a generic model without content features (G), a prompt-specific model with content features (PSWC), and a prompt-specific model without content features (PSNC). The evaluation of their method was carried out by sampling up to 3,000 essays for each prompt from a large number of prompts (113 issue prompts and 139 argument prompts). Each prompt has separate training and validation sets with 500 essays used as the training set. They built the generic model by using all essays from all prompts in training, except for the prompt being evaluated. Conversely, they built the prompt-specific model by using only the training data from the prompt being evaluated.

Their finding shows that the PSWC model only achieves a slight improvement over the PSNC and G models when evaluated using the quadratic weighted kappa (QWK) metric. The G model however has a higher discrepancy with the human scores. The study concluded that using prompt-specific features might not be so important since the PSWC model only achieves a slight improvement compared to the G and PSNC models. Also, the study found that the scores of the G and PSNC models are very similar, which supports the use of generic models

The problem with this study is that they used a large amount of training data for their generic model which came from 251 different prompts, which are usually not available in general. Moreover, their data were all collected from their GRE and TOEFL examinations. As such, the essays were more similar to each other across prompts compared to the essays from other datasets such as the ASAP dataset that have different score ranges and grading rubrics.

3.1.2 The ASAP Generic Model

The work of (Chen and He, 2013) also briefly explored the feasibility of a generic model. They tried performing a 5-fold cross validation on the ASAP dataset, combining the training and test data from different prompts, and created a generic model that works across all the prompts. The SVM model trained for classification and regression has a huge drop in the QWK score in this setting compared to the prompt-specific setting, while the rank-based approach only has a slight drop in the QWK score. However, their work does not investigate whether their generic model can perform well on a prompt that is not seen in the training data.

3.1.3 Task-Independent Features

(Zesch, Wojatzki, and Scholten-Akoun, 2015) tried to tackle the prompt-specificity problem by using task-independent features. They manually classified some of the commonly used features in AES into 2 groups: weakly task-dependent features and strongly task-dependent features.

They classified the following features as weakly task-dependent:

- Length features
- Syntax features
- Task-similarity features
- Set-dependent features

They classify the following features as strongly task-dependent:

- Occurrence features (comma, quotation, or exclamation mark)
- Style features
- Cohesion features

- Coherence features
- Error features
- Readability features

They created 2 AES systems, a full model using all the features and a reduced model using only the weakly task-dependent features. They performed an experiment on the ASAP dataset using both models, comparing the results of the model when trained and tested on the same prompt with the results of the model when trained and tested on a different prompt. They found that both models usually have a drop in performance when trained and tested on a different prompt. However, the reduced model has a smaller average drop in performance compared to the full model.

They also noted the difference between genres. They classified the ASAP prompts 3–6 as opinion tasks and 1, 2, 7, and 8 as source-based tasks. They noted that transfer within tasks has a lower average drop in performance compared to the drop between tasks.

3.2 Domain Adaptation

The knowledge learned from a single domain might not be directly applicable to another domain. For example, a named-entity recognizer 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, McDonald, and Pereira, 2006).

The approaches of domain adaptation can be divided into two categories based on the availability of labeled target data. The case where a small number of labeled target data is available is referred to as *supervised* domain adaptation (such as EasyAdapt and IW). The case where no labeled target domain data is available is referred to as *unsupervised* domain adaptation (such as SCL). Our work focuses 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 as word segmentation (Monroe, Green, and Manning, 2014), machine translation (Green, Cer, and Manning, 2014), and word sense disambiguation (Zhong, Ng, and Chan, 2008).

(Jiang and Zhai, 2007) proposed an instance weighting approach for domain adaptation. They weight training instances from the source and target domain differently. Training instances from the target domain are weighted more than the instances from the source domain. For each feature vector $\boldsymbol{x_i^s}$ and label $\boldsymbol{y_i^s}$ from the source domain, they introduce a parameter α_i indicating how close the conditional probability of the source domain $P_s(\boldsymbol{y_i^s}|\boldsymbol{x_i^s})$ is to that of the target domain $P_t(\boldsymbol{y_i^s}|\boldsymbol{x_i^s})$. Large α_i means that $P_t(\boldsymbol{y_i^s}|\boldsymbol{x_i^s})$ is similar to $P_s(\boldsymbol{y_i^s}|\boldsymbol{x_i^s})$ and small

α_i means the opposite. This parameter α_i will be used as a weight for the instances in the source domain. Instance pruning is a form of instance weighting where we remove "misleading" training instances for which $P_s(\boldsymbol{y_i^s}|\boldsymbol{x_i^s})$ is very different from $P_t(\boldsymbol{y_i^s}|\boldsymbol{x_i^s})$.

(Blitzer, McDonald, and Pereira, 2006) introduced an unsupervised domain adaptation method called structural correspondence learning (SCL). They focus on learning a common feature representation using unlabeled data from both source and target domains. SCL relies heavily on pivot features, features which behave the same way in both source and target domains. Using the pivot features, SCL learns a mapping θ that maps the original feature space to a lower dimensional shared representation of source and target domain features. This lower dimensional feature representation is then used as the input to a learning algorithm.

For each pivot feature, SCL will make a binary classification problem which predicts whether the feature exists in the text data. Then, it runs a linear classifier for each problem and combined their learned weights into a single matrix W. Finally, it uses Singular Value Decomposition (SVD) on W to get θ. Algorithm 1 shows the SCL algorithm.

Algorithm 1 The SCL algorithm.

Input: labeled source data, unlabeled source data, and unlabeled target data.

Output: a mapping θ.

1. Choose m pivot features. Run a linear classifier independently for each pivot and get their weight matrices $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m$.

2. Create a combined matrix $W = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m]$.

3. Perform SVD on W: $UDV^T = \text{SVD}(W)$.

4. Return $\theta = U^T_{[1:h,:]}$

The difference between prompts can be modeled as a domain adaptation problem. In this case, the domains will be the prompts, since AES systems are usually trained for a single prompt. One prompt will act as the source domain for which we have a lot of data. The other prompt will act as the target domain for which we have minimal amount of data. By doing this, we can transfer the knowledge we learn across prompts. In our work (Phandi, Chai, and Ng, 2015), we achieve domain adaptation by using Correlated Bayesian Linear Ridge Regression. Our work has been published in the EMNLP 2015 conference.

Chapter 4

Correlated Bayesian Linear Ridge Regression

We propose a novel domain adaptation technique based on Bayesian linear ridge regression in order to build an automated essay scoring system that works well on new essay prompts. We choose Bayesian linear ridge regression because it is simple and robust enough for automated essay scoring.

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

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

Here, α and λ are Gamma distributed hyper-parameters of the model; α_1 and λ_1 are the shape parameters for the Gamma distribution of α and λ respectively; α_2 and λ_2 are the inverse scale parameter for the Gamma distribution of α and λ respectively; $\boldsymbol{w} \in \mathbb{R}^p$ 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 respectively. 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

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

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 target and the source tasks, 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,

$$\mathbb{E}(f^{\mathrm{t}}(\boldsymbol{x}) f^{\mathrm{s}}(\boldsymbol{x}')) = \lambda^{-1} \rho \boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}'. \tag{4.1}$$

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 traverse 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

$$\begin{aligned} \Phi^{\mathrm{t}}(\boldsymbol{x}) &= \langle \boldsymbol{x}, \mathbf{0}_p \rangle \,, \\ \Phi^{\mathrm{s}}(\boldsymbol{x}) &= \left\langle \rho \boldsymbol{x}, (1-\rho^2)^{1/2} \boldsymbol{x} \right\rangle \end{aligned} \tag{4.2}$$

in $\mathbb{R}^{2p}$ for the target and the source tasks. Associated with this expanded feature representation is the weight vector $\boldsymbol{w} \stackrel{\text{def}}{=} (\boldsymbol{w}^{\mathrm{t}}, \boldsymbol{w}^{\mathrm{s}})$ 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, Cohn, and Specia, 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 (4.1). Other choices are possible, as long as equation (4.1) 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 domain prompts in this essay scoring task is improbable. In our application, we slightly bias the correlations towards zero with $a = 1/10$[4] 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 (4.2) is

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

Similarly, let $\boldsymbol{y}$ be the stacking of $\boldsymbol{y}^{\mathrm{t}}$ and $\boldsymbol{y}^{\mathrm{s}}$. Let $K \stackrel{\text{def}}{=} \lambda^{-1} X X^{\mathrm{T}} + \alpha^{-1} I$, 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

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

[4] We set a to be $1/10$ but other small values will also work.

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

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

where $\gamma \stackrel{\text{def}}{=} K^{-1}y$ 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 $\boldsymbol{x}_*^{\mathrm{T}} \bar{\boldsymbol{w}}^{\mathrm{t}}$.

Chapter 5

Experiments

In this chapter, 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[5] for our domain adaptation experiments. This dataset contains 8 prompts of different genres. All the essays were graded by at least 2 human graders. Details of this dataset are shown in Table 1.1 in Section 1.2.3.

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 $1 \rightarrow 2$, $3 \rightarrow 4$, $5 \rightarrow 6$, and $7 \rightarrow 8$, where the pair $1 \rightarrow 2$ 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]https://www.kaggle.com/c/asap-aes/data

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.

Our system uses EASE features together with Bayesian linear ridge regression (BLRR). We choose BLRR as our learning algorithm so as to use the correlated BLRR approach. EASE is created by one of the winners of the ASAP competition so the features they use have been proven to be robust. We perform the calculation of useful n-grams features 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.

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, 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 Same as SharedHyper, but with $\rho = 0.5$;

Concat Same as SharedHyper, but with $\rho = 1.0$;

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}.

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 suitable for essay scoring since it takes into account the agreement that occurrs by chance. 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, and E are the matrices of observed frequencies, weights, and expected frequencies respectively. Matrix entry $O_{i,j}$ 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

		B			**Total**
		1	**2**	**3**	
A	**1**	30	7	13	50
	2	5	20	5	30
	3	5	3	12	20
Total		40	30	30	

Table 5.1: Example matrix O of observed frequencies

		B			**Total**
		1	**2**	**3**	
A	**1**	20	15	15	50
	2	12	9	9	30
	3	8	6	6	20
Total		40	30	30	

Table 5.2: Example matrix E of expected frequencies

scores. Matrix E is calculated by taking the outer product between the frequency vectors of the two raters, which are then normalized to have the same sum as O.

We will illustrate the calculation of QWK by an example. Assuming we have 2 raters, A and B, with the score range of 1–3. Table 5.1 shows an example observed matrix O that we might have in this situation. From this matrix, we can take the normalized outer product to produce the expected matrix E as shown in Table 5.2. For example, $E_{1,1}$ is computed by multiplying the total A_1 (50) with the total B_1 (40) and dividing the product by the total number of essays to normalize it: $E_{1,1} = \frac{50*40}{100} = 20$. We calculate the weight matrix w as shown in Table 5.3. For example, $w_{1,2} = \frac{(1-2)^2}{(3-1)^2} = \frac{1}{4} = 0.25$ (note that the number of possible scores N is 3 (1, 2, and 3)). Finally, we calculate the QWK, $\kappa = 1 - \frac{0.25\times(7+5+5+3)+1\times(13+5)}{0.25\times(15+12+9+6)+1\times(15+8)} = 0.31$.

		B		
		1	**2**	**3**
A	**1**	0	0.25	1
	2	0.25	0	0.25
	3	1	0.25	0

Table 5.3: Example weight matrix W

Chapter 6

Results and Discussion

In-domain results for comparison First, we determine indicative upper bounds on the QWK scores using 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 6.1. We see that the BLRR scores are close to the the human agreement scores for prompt 1 and 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 6.1 for in-domain scores.

Importance of domain adaptation The results of the domain adaptation experiments are tabulated in Table 6.2, where the best scores are bold-faced and the second-best scores are underlined. As expected, for pairs $1 \rightarrow 2$, $3 \rightarrow 4$, and $5 \rightarrow 6$, all the scores are below their corresponding upper bounds from the in-domain set-

	QWK scores		
Prompt #	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
Average	0.704	0.699	0.754

Table 6.1: In-domain experimental results.

ting in Table 6.1. However, for pair $7 \rightarrow 8$, 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 1.1. 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 6.2 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,

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

Table 6.2: 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 bold-faced and the second-best score is underlined.

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 $5 \rightarrow 6$ and $7 \rightarrow 8$ 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 2.1 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. Figure 6.1 shows the average ρ estimated by ML-ρ over five folds. 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.

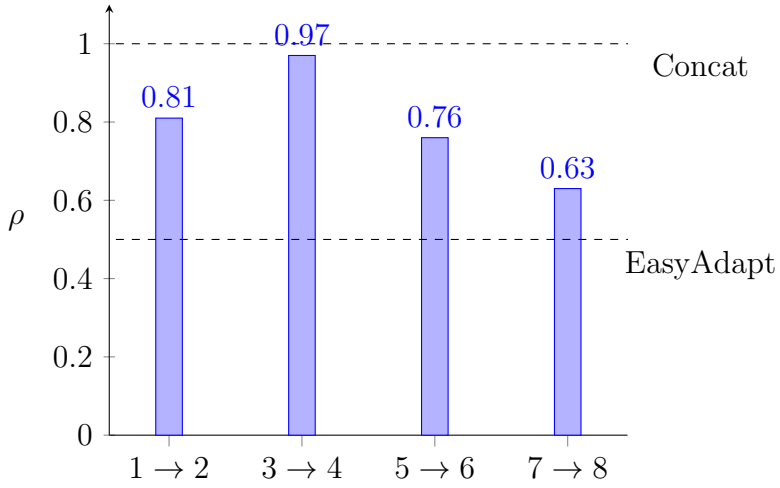

Figure 6.1: The average ρ value estimated by ML-ρ

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 choosing the prior p_ρ on ρ to better reflect the nature of the essays involved.

Chapter 7

Conclusion

In this thesis, we first gave an overview of the Automated Essay Scoring (AES) task, including various approaches to AES proposed in the literature. We identified a prompt-specificity problem on current AES systems and proposed domain adaptation as one way to solve the problem. We have shown that domain adaptation can achieve better results compared to using just the small number of target domain essays or just using a large number of essays from a different domain. We propose a novel domain adaptation technique based on Bayesian linear ridge regression and show its effectiveness on the ASAP dataset. This is the main contribution of this thesis.

There is some future work that can be done in the field of AES. An important topic is addressing the question of whether an AES system can be gamed. We can investigate how the current AES systems can be tricked and see how to make it more robust to prevent it from being gamed. One particular concern is that most AES systems use the length feature due to its high correlation with the human-assigned score. This feature can be easily gamed by writing a long but nonsensical essay. A preliminary study has been done by ETS on their e-rater system (Powers et al., 2001).

Another possible direction is to continue looking at the prompt specificity problem. ETS has proposed building a generic model from their data. It would be interesting to see whether their method will work on a dataset with a more diverse set of prompts such as the ASAP dataset. We can also experiment with genre-based domain adaptation to see whether it will help even more compared to prompt-based domain adaptation.

References

Attali, Yigal, Brent Bridgeman, and Catherine Trapani. 2010. Performance of a generic approach in automated essay scoring. *The Journal of Technology, Learning and Assessment*, (3).

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

Beck, Daniel, 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*.

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

Blitzer, John, 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*.

Breland, Hunter M, Robert J Jones, Laura Jenkins, Marion Paynter, Judith Pollack, and Y Fai Fong. 1994. The college board vocabulary study. *ETS Research Report Series*.

Burstein, Jill, Joel Tetreault, and Slava Andreyev. 2010. Using entity-based features to model coherence in student essays. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Callan, James P, W Bruce Croft, and Stephen M Harding. 1992. The Inquery retrieval system. In *Database and Expert Systems Applications*. Springer.

Chen, Hongbo 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*.

Connors, Robert J. 1981. The rise and fall of the modes of discourse. *College Composition and Communication*.

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

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

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

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

Granger, Sylviane, Estelle Dagneaux, Fanny Meunier, and Magali Paquot. 2009. The International Corpus of Learner English. version 2. Handbook and CD-ROM.

Green, Spence, 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*.

Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The Weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*.

Higgins, Derrick, Jill Burstein, Daniel Marcu, and Claudia Gentile. 2004. Evaluating multiple aspects of coherence in student essays. In *Proceedings of Human Language Technologies: The 2004 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

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

Joachims, T. 1999. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, MA.

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

Landauer, Thomas K, Darrell Laham, and Peter W Foltz. 2003. Automated scoring and annotation of essays with the intelligent essay assessor. *Automated essay scoring: A cross-disciplinary perspective*, pages 87–112.

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

Miltsakaki, Eleni and Karen Kukich. 2004. Evaluation of text coherence for electronic essay scoring systems. *Natural Language Engineering*.

Monroe, Will, 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*.

Page, Ellis B, John P Poggio, and Timothy Z Keith. 1997. Computer analysis of student essays: Finding trait differences in student profile. In *Proceedings of the 1997 Annual Meeting of the American Educational Research Association*.

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

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011.

Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*.

Persing, Isaac, Alan Davis, and Vincent Ng. 2010. Modeling organization in student essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Persing, Isaac and Vincent Ng. 2013. Modeling thesis clarity in student essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

Persing, Isaac and Vincent Ng. 2014. Modeling prompt adherence in student essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Persing, Isaac and Vincent Ng. 2015. Modeling argument strength in student essays. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

Phandi, Peter, Kian Ming A Chai, and Hwee Tou Ng. 2015. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Powers, Donald E, Jill C Burstein, Martin Chodorow, Mary E Fowles, and Karen Kukich. 2001. Stumping e-rater: Challenging the validity of automated essay scoring. *ETS Research Report Series*.

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

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

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

Shermis, Mark D, Howard R Mzumara, Jennifer Olson, and Susanmarie Harrington. 2001. On-line grading of student essays: PEG goes on the world wide web. *Assessment & Evaluation in Higher Education*.

Wu, Qiang, Chris JC Burges, Krysta M Svore, and Jianfeng Gao. 2008. Ranking, boosting, and model adaptation. Technical report, Microsoft.

Yannakoudakis, Helen and Ted Briscoe. 2012. Modeling coherence in ESOL learner texts. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics.

Yannakoudakis, Helen, 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*.

Zesch, Torsten, Michael Wojatzki, and Dirk Scholten-Akoun. 2015. Task-independent features for automated essay grading. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*.

Zhong, Zhi, 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*.

Appendix A

ASAP Prompts

Prompt 1

More and more people use computers, but not everyone agrees that this benefits society. Those who support advances in technology believe that computers have a positive effect on people. They teach hand-eye coordination, give people the ability to learn about faraway places and people, and even allow people to talk online with other people. Others have different ideas. Some experts are concerned that people are spending too much time on their computers and less time exercising, enjoying nature, and interacting with family and friends.

Write a letter to your local newspaper in which you state your opinion on the effects computers have on people. Persuade the readers to agree with you.

Prompt 2

Censorship in the Libraries "All of us can think of a book that we hope none of our children or any other children have taken off the shelf. But if I have the right to remove that book from the shelf – that work I abhor – then you

also have exactly the same right and so does everyone else. And then we have no books left on the shelf for any of us." –Katherine Paterson, Author

Write a persuasive essay to a newspaper reflecting your vies on censorship in libraries. Do you believe that certain materials, such as books, music, movies, magazines, etc., should be removed from the shelves if they are found offensive? Support your position with convincing arguments from your own experience, observations, and/or reading.

Prompt 3

Write a response that explains how the features of the setting affect the cyclist. In your response, include examples from the essay that support your conclusion.

Prompt 4

Read the last paragraph of the story.

"When they come back, Saeng vowed silently to herself, in the spring, when the snows melt and the geese return and this hibiscus is budding, then I will take that test again."

Write a response that explains why the author concludes the story with this paragraph. In your response, include details and examples from the story that support your ideas.

Prompt 5

Describe the mood created by the author in the memoir. Support your answer with relevant and specific information from the memoir.

Prompt 6

Based on the excerpt, describe the obstacles the builders of the Empire State Building faced in attempting to allow dirigibles to dock there. Support your answer with relevant and specific information from the excerpt.

Prompt 7

Write about patience. Being patient means that you are understanding and tolerant. A patient person experience difficulties without complaining.

Do only one of the following: write a story about a time when you were patient OR write a story about a time when someone you know was patient OR write a story in your own way about patience.

Prompt 8

We all understand the benefits of laughter. For example, someone once said, Laughter is the shortest distance between two people. Many other people believe that laughter is an important part of any relationship. Tell a true story in which laughter was one element or part.