Constrained Multi-Task Learning for Automated Essay Scoring

Ronan Cummins
ALTA Institute
Computer Lab
University of Cambridge

Meng Zhang
ALTA Institute
Computer Lab
University of Cambridge

Ted Briscoe
ALTA Institute
Computer Lab
University of Cambridge

{rc635,mz342,ejb}@cl.cam.ac.uk

Abstract

Supervised machine learning models for automated essay scoring (AES) usually require substantial task-specific training data in order to make accurate predictions for a particular writing task. This limitation hinders their utility, and consequently their deployment in real-world settings. In this paper, we overcome this shortcoming using a constrained multi-task pairwise-preference learning approach that enables the data from multiple tasks to be combined effectively.

Furthermore, contrary to some recent research, we show that high performance AES systems can be built with little or no task-specific training data. We perform a detailed study of our approach on a publicly available dataset in scenarios where we have varying amounts of task-specific training data and in scenarios where the number of tasks increases.

1 Introduction

Automated essay scoring (AES) involves the prediction of a score (or scores) relating to the *quality* of an extended piece of written text (Page, 1966). With the burden involved in manually grading student texts and the increase in the number of ESL (English as a second language) learners worldwide, research into AES is increasingly seen as playing a viable role in assessment. Automating the assessment process is not only useful for educators but also for learners, as it can provide instant feedback and encourage iterative refinement of their writing.

The AES task has usually been addressed using machine learning. Given a set of texts and associated gold scores, machine learning approaches aim to build models that can generalise to unseen instances. Regression (Page, 1994; Persing and Ng, 2014; Phandi et al., 2015), classification (Larkey, 1998; Rudner and Liang, 2002), and preference-ranking[1] approaches (Yannakoudakis et al., 2011) have all been applied to the task. In general, machine learning models only perform well when the training and test instances are from similar distributions. However, it is usually the case that essays are written in response to prompts which are carefully designed to elicit answers according to a number of dimensions (e.g. register, topic, and genre). For example, Table 1 shows extracts from two prompts from a publicly available dataset[2] that aim to elicit different genres of persuasive/argumentative responses on different topics.

Most previous work on AES has either ignored the differences between essays written in response to different prompts (Yannakoudakis et al., 2011) with the aim of building general AES systems, or has built prompt-specific models for each prompt independently (Chen and He, 2013; Persing and Ng, 2014). One of the problems hindering the wide-scale adoption and deployment of AES systems is the dependence on prompt-specific training data, i.e. substantial model retraining is often needed when a new prompt is released. Therefore, systems that can adapt to new writing tasks (i.e. prompts) with relatively few new task-specific training examples are particularly appealing. For example, a system that is trained using only responses from prompt #1 in Table 1 may not generalise well to essays written in response to prompt #2, and vice versa. Even more complications arise when the scoring scale, marking criteria, and/or grade level (i.e. educational stage) vary from task

[1] also known as pairwise learning-to-rank

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

#1	*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.*
#2	*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.*

Table 1: Two sample writing tasks from the ASAP (Automated Student Assessment Prize) dataset.

to task. If essays written in response to different tasks are marked on different scoring scales, then the actual scores assigned to essays across tasks are not directly comparable. This effect becomes even more pronounced when prompts are aimed at students in different educational stages.

In this paper, we address this problem of prompt adaptation using multi-task learning. In particular, we treat each prompt as a different task and introduce a constrained preference-ranking approach that can learn from multiple tasks even when the scoring scale and marking criteria are different across tasks. Our constrained preference-ranking approach significantly increases performance over a strong baseline system when there is limited prompt-specific training data available. Furthermore, we perform a detailed study using varying amounts of task-specific training data and varying numbers of tasks. First, we review some related work.

2 Related Work

A number of commercially available systems for AES, have been developed using machine learning techniques. These include PEG (Project Essay Grade) (Page, 2003), e-Rater (Attali and Burstein, 2006), and Intelligent Essay Assessor (IEA) (Landauer et al., 1998). Beyond commercial systems, there has been much research into varying aspects involved in automated assessment, including coherence (Higgins et al., 2004; Yannakoudakis and Briscoe, 2012), prompt-relevance (Persing and Ng, 2014; Higgins et al., 2006), argumentation (Labeke et al., 2013; Somasundaran et al., 2014; Persing and Ng, 2015), grammatical error detection and correction (Rozovskaya and Roth, 2011; Felice et al., 2014), and the development of publicly available resources (Yannakoudakis et al., 2011; Dahlmeier et al., 2013; Persing and Ng, 2014; Ng et al., 2014).

While most of the early commercially available systems use linear-regression models to map essay features to a score, a number of more sophisticated approaches have been developed. Preference-ranking (or pairwise learning-to-rank) has been shown to outperform regression for the AES problem (Yannakoudakis et al., 2011). However, they did not study prompt-specific models, as their models used training data originating from different prompts. We also adopt a preference-ranking approach but explicitly model prompt effects during learning. Algorithms that aim to directly maximise an evaluation metric have also been attempted. A listwise learning-to-rank approach (Chen and He, 2013) that directly optimises quadratic-weighted Kappa, a commonly used evaluation measure in AES, has also shown promising results.

Using training data from natural language tasks to boost performance of related tasks, for which there is limited training data, has received much attention of late (Collobert and Weston, 2008; Duh et al., 2010; Cheng et al., 2015). However, there have been relatively few attempts to apply transfer learning to automated assessment tasks. Notwithstanding, Napoles and Callison-Burch (2015) use a multi-task approach to model differences in assessors, while Heilman and Madnani (2013) specifically focus on domain-adaptation for short answer scoring over common scales. Most relevant is the work of Phandi et al. (2015), who applied domain-adaptation to the AES task using EasyAdapt (EA) (Daume III, 2007). They showed that supplementing a Bayesian linear ridge regression model (BLRR) with data from one other source domain is beneficial when there is limited target domain data. However, it was shown that simply using the source domain data as extra training data outperformed the EA domain adaptation approach in three out of four cases. One major limitation to their approach was that in many instances the source domain and target domain pairs were from different grade levels. This means that any attempt to resolve scores to a common scale is undermined by the fact that the gold scores are not comparable across domains, as the essays were written by students of different educational levels. A further limitation is that multi-domain adapta-

tion (whereby one has access to multiple source domains) was not considered.

The main difference between our work and previous work is that our model incorporates *multiple* source tasks and introduces a learning mechanism that enables us to combine these tasks even when the scores across tasks are not directly comparable. This has not been achieved before. This is non-trivial as it is difficult to see how this can be accomplished using a standard linear-regression approach. Furthermore, we perform the first comprehensive study of multi-task learning for AES using different training set sizes for a number of different learning scenarios.

3 Preference Ranking Model

In this section, we describe our baseline AES model which is somewhat similar to that developed by Yannakoudakis et al. (2011).

3.1 Perceptron Ranking (TAP$_{rank}$)

We use a preference-ranking model based on a binary margin-based linear classifier (the Timed Aggregate Perceptron or TAP) (Briscoe et al., 2010). In its simplest form this Perceptron uses batch learning to learn a decision boundary for classifying an input vector $\boldsymbol{x_i}$ as belonging to one of two categories. A timing-variable τ (set to 1.0 by default) controls both the learning rate and the number of epochs during training. A preference-ranking model is then built by learning to classify pairwise *difference vectors*, i.e. learning a weight vector $\boldsymbol{w}$ such that $\boldsymbol{w}(\boldsymbol{x_i} - \boldsymbol{x_j}) > \delta$, when essay i has a higher gold score than essay j, where δ is the one-sided margin[3] (Joachims, 2002; Chapelle and Keerthi, 2010). Therefore, instead of directly learning to predict the gold score of an essay vector, the model learns a weight vector $\boldsymbol{w}$ that minimizes the misclassification of *difference vectors*. Given that the number of pairwise difference vectors in a moderately sized dataset can be extremely large, the training set is reduced by randomly sampling difference vectors according to a user-defined probability (Briscoe et al., 2010). In all experiments in our paper we choose this probability such that $5n$ difference vectors are sampled, where n is the number of training instances (essays) used. We did not tune any of the hyper-parameters of the model.

[3]This margin is set to $\delta = 2.0$ by default.

3.2 From Rankings to Predicted Scores

As the weight vector $\boldsymbol{w}$ is optimized for pairwise ranking, a further step is needed to use the ranking model for predicting a score. In particular, for each of the n vectors in our training set, a real-scalar value is assigned according to the dot-product of the weight vector and the training instance (i.e. $\boldsymbol{w} \cdot \boldsymbol{x_i}$), essentially giving its distance (or margin) from the zero vector. Then using the training data, we train a one-dimensional linear regression model $\beta + \epsilon$ to map these assignments to the gold score of each instance.

Finally, to make a prediction $\hat{y}$ for a test vector, we first calculate its distance from the zero vector using $\boldsymbol{w} \cdot \boldsymbol{x_i}$ and map it to the scoring scale using the linear regression model $\hat{y} = \beta(\boldsymbol{w} \cdot \boldsymbol{x_i}) + \epsilon$. For brevity we denote this entire approach (a ranking and a linear regression step) to predicting the final score as **TAP**.

3.3 Features

The set of features used for our ranking model is similar to those identified in previous work (Yannakoudakis et al., 2011; Phandi et al., 2015) and is as follows:

1. word unigrams, bigrams, and trigrams
2. POS (part-of-speech) counts
3. essay length (as the number of unique words)
4. GRs (grammatical relations)
5. max-word length and min-sentence length
6. the presence of cohesive devices
7. an estimated error rate

Each essay is processed by the RASP system (Briscoe et al., 2006) with the standard tokenisation and sentence boundary detection modules. All n-grams are extracted from the tokenised sentences. The grammatical relations (GRs) are extracted from the top parse of each sentence in the essay. The presence of cohesive devices are used as features. In particular, we use four categories (i.e. addition, comparison, contrast and conclusion) which are hypothesised to measure the cohesion of a text.

The error rate is estimated based on a language model using ukWaC (Ferraresi et al., 2008) which contains more than 2 billion English tokens. A trigram in an essay will be treated as an error if it

Details						System Performance (QW-κ)		
Task	**# essays**	**Grade Level**	**Original Scale**	**Mean Score Resolved (0-60)**	**Human Agreement**	**BLRR Phandi**	**SVM Phandi**	**TAP**
1	1783	8	2-12	39	0.721	0.761	0.781	**0.815**
2	1800	10	1-6	29	0.814	0.606	0.621	**0.674**
3	1726	10	0-3	37	0.769	0.621	0.630	**0.642**
4	1772	10	0-3	29	0.851	0.742	0.749	**0.789**
5	1805	8	0-4	36	0.753	0.784	0.782	**0.801**
6	1800	10	0-4	41	0.776	0.775	0.771	**0.793**
7	1569	7	0-30	32	0.721	0.730	0.727	**0.772**
8	723	10	0-60	37	0.629	0.617	0.534	**0.688**

Table 2: Details of ASAP dataset and a preliminary evaluation of the performance of our **TAP** baseline against previous work (Phandi et al., 2015). All models used only task-specific data and 5-fold cross-validation. Best result is in **bold**.

is not found in the language model. Spelling errors are detected using a dictionary lookup, while a rule-based error module (Andersen et al., 2013) with rules generated from the Cambridge Learner Corpus (CLC) (Nicholls, 2003) is used to detect further errors. Finally, the unigrams, bigrams and trigrams are weighted by *tf-idf* (Sparck Jones, 1972), while all other features are weighted by their actual frequency in the essay.

4 Data and Preliminary Evaluation

In order to compare our baseline with previous work, we use the ASAP (Automated Student Assessment Prize) public dataset. Some details of the essays for the eight tasks in the dataset are described in the Table 2. The prompts elicit responses of different genres and of different lengths. In particular, it is important to note that the prompts have different scoring scales and are associated with different grade levels (7-10). Furthermore, the gold scores are distributed differently even if resolved to a common 0-60 scale. In order to benchmark our baseline system against previously developed approaches (BLRR and SVM regression (Phandi et al., 2015)) which use this data, we learned task-specific models using 5-fold cross-validation within each of the eight ASAP sets and aim to predict the unresolved *original* score as per previous work. We present the quadratic weighted kappa (QW-κ) of the systems in Table 2.[4] Our baseline preference-ranking model (TAP) outperforms previous approaches on task-specific data. It is worth noting that we did not tune either of the hyperparameters of TAP.

[4]The results for BLRR and SVM regression are taken directly from the original work and it is unlikely that we have used the exact same fold split. Regardless, the consistent increases mean that TAP represents a strong baseline system upon which we develop our constrained multi-task approach.

5 Multi-Task Learning

For multi-task learning we use EA encoding (Daume III, 2007) extended over k tasks $\mathcal{T}_{j=1..k}$ where each essay x_i is associated with one task $x_i \in \mathcal{T}_j$. The transfer-learning algorithm takes a set of input vectors associated with the essays, and for each vector $\boldsymbol{x}_i \in \mathbb{R}^F$ maps it via $\Phi(\boldsymbol{x}_i)$ to a higher dimensional space $\Phi(\boldsymbol{x}_i) \in \mathbb{R}^{(1+k)\cdot F}$. The encoding function $\Phi(\boldsymbol{x}_i)$ is as follows:

$$\Phi(\boldsymbol{x}) = \bigoplus_{j=0}^{k} f(\boldsymbol{x}, j) \tag{1}$$

where $\bigoplus$ is vector concatenation and $f(\boldsymbol{x}, j)$ is as follows:

$$f(\boldsymbol{x}, j) = \begin{cases} \boldsymbol{x}, & \text{if } j = 0 \\ \boldsymbol{x}, & \text{if } x \in \mathcal{T}_j \\ \mathbf{0}^F, & \text{otherwise} \end{cases} \tag{2}$$

Essentially, the encoding makes a task-specific copy of the original feature space of dimensionality F to ensure that there is one shared-representation and one task-specific representation for each input vector (with a zero vector for all other tasks). This approach can be seen as a re-encoding of the input vectors and can be used with any vector-based learning algorithm. Fig. 1 (left) shows an example of the extended feature vectors for three tasks $\mathcal{T}_j$ on different scoring scales. Using only the shared-representation (in blue) as input vectors to a learning algorithm results in a standard approach which does not learn task-specific characteristics. However, using the full representation allows the learning algorithm to capture both general and task-specific characteristics jointly. This simple encoding technique is easy to implement and has been shown to be useful for a number of NLP tasks (Daume III, 2007).

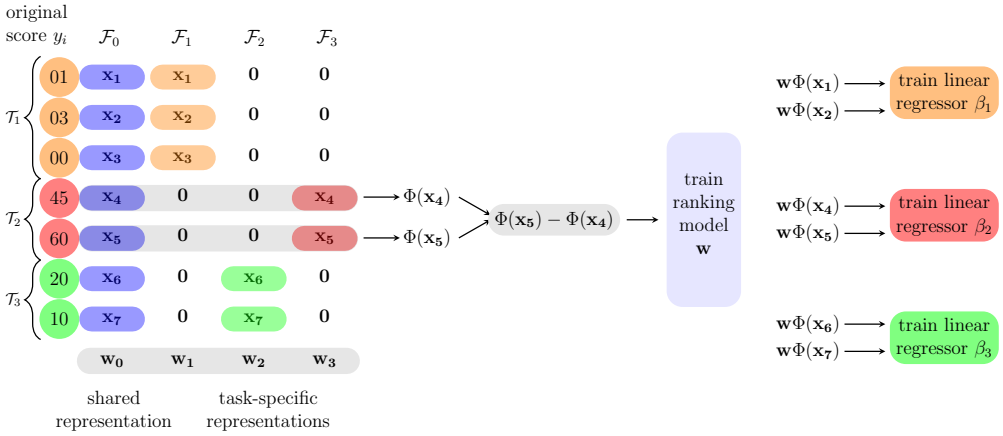

Figure 1: Example of the constrained multi-task learning approach for three tasks where the shared representation is in blue and the task-specific representations are in orange, red, and green. The original gold scores for each task $\mathcal{T}_j$ are on different scoring scales. The preference-ranking weight vector $\boldsymbol{w}$ to be learned is shown at the bottom. A one-dimensional linear regression model is learned for each task.

5.1 Constrained Preference-Ranking

Given essays from multiple tasks, it is often the case that the gold scores have different distributions, are not on the same scale, and have been marked using different criteria. Therefore, we introduce a modification to TAP (called cTAP$_{rank}$) that constrains the creation of pairwise difference vectors when training the weight vector $\boldsymbol{w}$. In particular, during training we ensure that pairwise difference vectors are not created from pairs of essays originating from different tasks.[5] We ensure that the same number of difference vectors are sampled during training for both TAP$_{rank}$ and our constrained version (i.e. both models use the same number of training instances). Figure 1 shows an example of the creation of a valid pairwise-difference vector in the multi-task framework.

Furthermore, for cTAP$_{rank}$ we train a final linear regression step on each of the task-specific training data separately. Therefore, we predict a score y for essay $\boldsymbol{x_i}$ for task $\mathcal{T}_j$ as $\hat{y} = \beta_j(\boldsymbol{w} \cdot \boldsymbol{x_i}) + \epsilon_j$. This is because for cTAP$_{rank}$ we assume that scores across tasks are not necessarily comparable. Therefore, although we utilise information originating from different tasks, the approach *never* mixes or directly compares instances originating from different tasks. This approach to predicting the final score is denoted **cTAP**.

[5]The same effect can be achieved in SVM^{rank} by encoding the prompt/task using the query id (*qid*). This constraint is analogous to the way SVM^{rank} is used in information retrieval where document relevance scores returned from different queries are not comparable.

6 Experimental Set-up

In this section, we outline the different learning scenarios, data folds, and evaluation metrics used in our main experiments.

6.1 Learning Approaches

We use the same features outlined in Section 3.3 to encode feature vectors for our learning approaches. In particular we study three learning approaches denoted and summarised as follows:

TAP: which uses the TAP$_{rank}$ algorithm with input vectors $\boldsymbol{x_i}$ of dimensionality F.

MTL-TAP: which uses the TAP$_{rank}$ algorithm with MTL extended input vectors $\Phi(\boldsymbol{x}_i)$.

MTL-cTAP: which uses the cTAP$_{rank}$ algorithm with MTL extended input vectors $\Phi(\boldsymbol{x}_i)$.[6]

For TAP and MTL-TAP, we attempt to resolve the essay score to a common scale (0-60) and subsequently train and test using this resolved scale. We then convert the score back to the original prompt-specific scale for evaluation. This is the approach used by the work most similar to ours (Phandi et al., 2015). It is worth noting that the resolution of scores to a common scale prior to training is *necessary* for both TAP and MTL-TAP when using data from multiple ASAP prompts. However, this step is *not* required for MTL-cTAP as this algorithm learns a ranking function $\boldsymbol{w}$ without directly comparing essays from different sets during training. Furthermore, the final regres-

[6]In the standard learning scenario when only target task data is available, MTL-TAP and MTL-cTAP are identical.

System	Target Task/Prompts 1	2	3	4	5	6	7	8
Tgt-TAP	0.830	0.728	0.717	0.842	0.851	0.811	0.790	0.730
Src-TAP	0.779	0.663	0.703	0.735	0.789	0.688	0.616	**0.625**
Src-MTL-TAP	0.824‡	0.683†	0.728‡	0.771‡	**0.829**‡	0.699	0.737‡	0.575
Src-MTL-cTAP	**0.826**‡	**0.698**$^{\star}_{\star}$‡	**0.729**‡	**0.773**$_{\star}$‡	0.827‡	**0.702**†$_{\star}$	**0.744**$^{\star}_{\star}$‡	0.589$^{\star}_{\star}$
All-TAP	0.806	0.652	0.702	0.805	0.814	0.802	0.728	0.629
All-MTL-TAP	0.831‡	0.722‡	0.728‡	0.823‡	0.849‡	0.808	0.783‡	0.680‡
All-MTL-cTAP	**0.832**‡	**0.731**‡$_{\star}$	**0.729**‡$_{\star}$	**0.840**$^{\star}_{\star}$‡	**0.852**‡$_{\star}$	**0.810**†	**0.802**$^{\star}_{\star}$‡	**0.717**$^{\star}_{\star}$‡

Table 3: Average Spearman ρ of systems over two-folds on the ASAP dataset. The best approach per prompt is in **bold**. ‡ (†) means that ρ is statistically greater than **Src-TAP** (top half) and **All-TAP** (bottom half) using the Steiger test at the 0.05 level (‡ means significant for both folds, † means for one of the folds), while $^{\star}_{\star}$ means statistically greater than **All-MTL-TAP** on both folds ($_{\star}$ for one fold).

System	Target Tasks/Prompts 1	2	3	4	5	6	7	8
Tgt-TAP	0.813	0.667	0.626	0.779	0.789	0.763	0.758	0.665
All-TAP	0.803	0.598	0.583	0.648	0.747	0.741	0.674	0.462
All-MTL-TAP	**0.825**‡	0.658‡	0.643‡	0.702‡	0.784‡	0.759‡	0.778‡	0.692‡
All-MTL-cTAP	0.816‡	**0.667**‡$_{\star}$	**0.654**$^{\star}_{\star}$‡	**0.783**$^{\star}_{\star}$‡	**0.801**$^{\star}_{\star}$‡	**0.778**$^{\star}_{\star}$‡	**0.787**‡$_{\star}$	**0.692**‡

Table 4: Average QW-κ of systems over two-folds on the ASAP dataset. The best approach per prompt is in **bold**. ‡ (†) means that κ is statistically ($p < 0.05$) greater than **All-TAP** using an approximate randomisation test (Yeh, 2000) using 50,000 samples. $^{\star}_{\star}$ means statistically greater than **All-MTL-TAP** on both folds ($_{\star}$ for one fold).

sion step in cTAP only uses original target task scores and therefore predicts scores on the correct scoring scale for the task. We study the three different learning approaches, TAP, MTL-TAP, and MTL-cTAP, in the following scenarios:

All: where the approach uses data from both the target task and the available source tasks.

Tgt: where the approach uses data from the target task only.

Src: where the approach uses data from only the available source tasks.

6.2 Data Folds

For our main experiments we divide the essays associated with each of the eight tasks into two folds. For *all* subsequent experiments, we train using data in one fold (often associated with multiple tasks) and test on data in the remaining fold of the specific target task. We report results for each task separately. These splits allow us to perform studies of all three learning approaches (TAP, MTL-TAP, and MTL-cTAP) using varying amounts of source and target task training data.

6.3 Evaluation Metrics

We use both Spearman's ρ correlation and Quadratic-weighted κ (QW-κ) to evaluate the performance of all approaches. Spearman's ρ measures the quality of the ranking of predicted scores produced by the system (i.e. the output from the ranking-preference model). We calculate Spearman's ρ using the ordinal gold score and the real-valued prediction on the original prompt-specific scoring scale of each prompt. Statistical significant differences between two correlations sharing one dependent variable (i.e. the gold scores) can be determined using Steiger's (1980) test.

QW-κ measures the chance corrected agreement between the predicted scores and the gold scores. QW-κ can be viewed as a measure of accuracy as it is lower when the predicted scores are further away from the gold scores. This metric measures both the quality of the ranking of scores and the quality of the linear regression step of our approach. These metrics are complementary as they measure different aspects of performance. We calculate QW-κ using the ordinal gold score and the real-valued prediction rounded to the nearest score on the original prompt-specific scale (see Table 2).

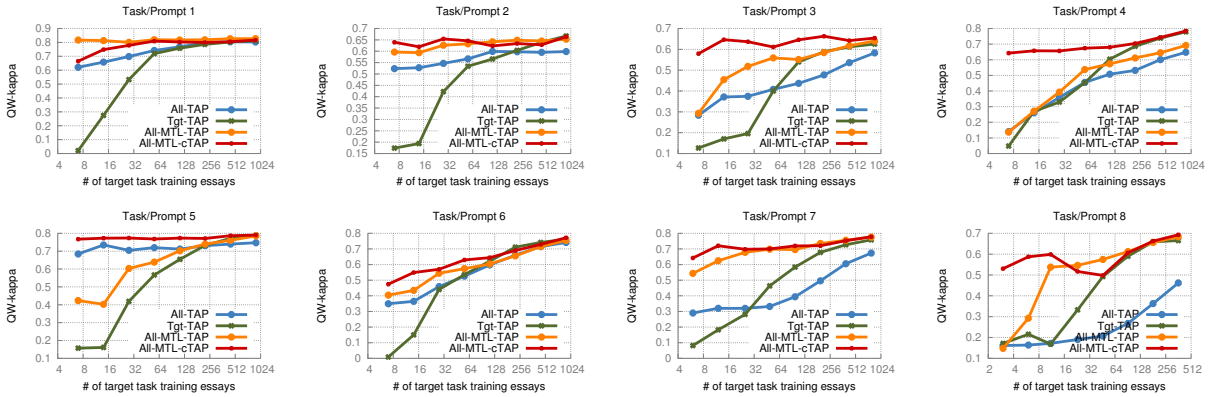

Figure 2: Average QW-κ over two folds for all tasks as size of target task training data increases

7 Results and Discussion

Table 3 and Table 4 show the performance of a number of models for both ρ and κ respectively. In general, we see that the **MTL** versions nearly always outperform the baseline TAP when using the same training data. This shows that multi-task learning is superior to simply using the source tasks as extra training data for the AES task. Interestingly this has not been shown before. Furthermore, the MTL-cTAP approach tends to be significantly better than the other for many prompts under varying scenarios for both Spearman's ρ and QW-κ. This shows that models that attempt to directly compare essays scores across certain writing-tasks lead to poorer performance.

When looking at Spearman's ρ in Table 3 we see that the models that do not use any target task data during training (Src) can achieve a performance which is close to the baseline that only uses all of the available target data (Tgt-TAP). This indicates that our system can *rank* essays well without any target task data. However, it is worth noting that *without* any target task training data and lacking any prior information as to the distribution of gold scores for the target task, achieving a consistently high accuracy (i.e. QW-κ) is extremely difficult (if not impossible). Therefore, Table 4 only shows results for models that make use of target task data.

For the models trained with data from all eight tasks, we can see that All-MTL-cTAP outperforms both All-TAP and All-MTL-TAP on most of the tasks for both evaluation metrics (ρ and κ). Interestingly, All-MTL-cTAP also outperforms Tgt-TAP on most of the prompts for both evaluation metrics. This indicates that All-MTL-cTAP manages to successfully incorporate useful information from the source tasks even when there is ample target-task data. We next look at scenarios when target-task training data is lacking.

7.1 Study of Target-Task Training Size

In real-world scenarios, it is often the case that we lack training data for a new writing task. We now report the results of an experiment that uses varying amounts of target-task training data. In particular, we use all source tasks and initially a small sample of task-specific data for each task (every 128^{th} target essay) and measure the performance of Tgt-TAP and the **All-*** models. We then double the amount of target-task training data used (by using every 64^{th} essay) and again measure performance, repeating this process until all target-task data is used. Figure 2 shows the performance of Tgt-TAP and the **All-*** models as target-task data increases.

In particular, Figure 2 shows that All-MTL-cTAP consistently outperforms all approaches in terms of agreement (QW-κ) and is particularly superior when there is very little target-task training data. It is worth remembering that All-MTL-cTAP only uses the target-task training instances for the final linear regression step. These results indicate that because the preference-ranking model performs so well, only a few target-task training instances are needed for the linear-regression step of All-MTL-cTAP. On the other hand, All-MTL-TAP uses all of the training instances in its final linear regression step, and performs significantly worse on a number of prompts. Again this shows the strengths of the constrained multi-task approach.

7.2 Study of Number of Source-tasks

All previous experiments that used source task data used the entire seven additional tasks. We

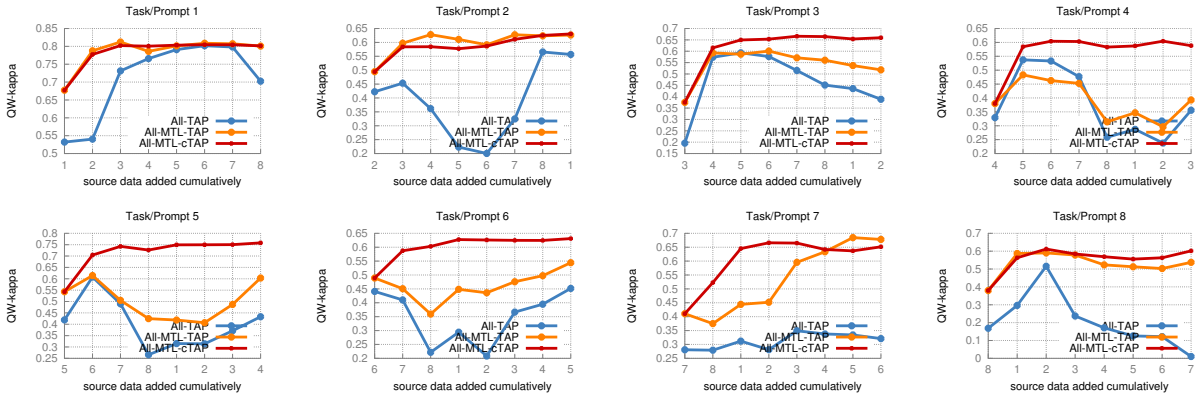

Figure 3: Average QW-κ over two folds as number of source tasks increases (using 25 target task instances)

now study the performance of the approaches as the number of source tasks changes. In particular, we limit the number of target task training instances to 25 and cumulatively add entire source task data in the order in which they occur in Table 2, starting with the source task appearing directly after the target task. We then measure performance at each stage. At the end of the process, each approach has access to all source tasks and the limited target task data.

Figure 3 shows the QW-κ for each prompt as the number of source tasks increases. We can see that All-TAP is the worst performing approach and often decreases as certain tasks are added as training data. All-MTL-cTAP is the best performing approach for nearly all prompts. Furthermore, All-MTL-cTAP is more robust than other approaches, as it rarely decreases in performance as the number of tasks increases.

8 Qualitative Analysis

As an indication of the type of interpretable information contained in the task-specific representations of the All-MTL-cTAP model, we examined the shared representation and two task-specific representations that relate to the example tasks outlined in Table 1. Table 5 shows the top weighted lexical features (i.e. unigrams, bigrams, or trigrams) (and their respective weights) in different parts of the All-MTL-cTAP model.

In general, we can see that the task-specific lexical components of the model capture topical aspects of the tasks and enable domain adaptation to occur. For example, we can see that *books*, *materials*, and *censorship* are highly discriminative lexical features for ranking essays written in response to task #2. The shared representation contains highly weighted lexical features across all tasks and captures vocabulary items useful for ranking in general. While this analysis gives us some insight into our model, it is more difficult to interpret the weights of other feature types (e.g. POS, GRs) across different parts of the model. We leave further analysis of our approach to future work.

9 Discussion and Conclusion

Unlike previous work (Phandi et al., 2015) we have shown, for the first time, that MTL outperforms an approach of simply using source task data as extra training data. This is because our approach uses information from multiple tasks without directly relying on the comparability of gold scores across tasks. Furthermore, it was concluded in previous work that at least some target-task training data is necessary to build high performing AES systems. However, as seen in Table 3, high performance rankers (ρ) can be built *without any* target-task data. Nevertheless, it is worth noting that without any target-data, accurately predicting the actual score (high κ) is extremely difficult. Therefore, although some extra information (i.e. the expected distribution of gold scores) would need to be used to produce accurate scores with a high quality ranker, the ranking is still useful for assessment in a number of scenarios (e.g. grading on a curve where the distribution of student scores is predefined).

The main approach adopted in this paper is quite similar to using SVM^{rank} (Joachims, 2002) while encoding the prompt id as the qid. When combined with a multi-task learning technique this allows the preference-ranking algorithm to learn

Shared		Task #1		Task #2	
2.024	offensive	1.146	this	2.027	offensive
1.852	hydrogen	0.985	less	1.229	books
1.641	hibiscus	0.980	computers	0.764	do_n't
1.602	shows	0.673	very	0.720	materials
1.357	strong	0.661	would	0.680	censorship
1.326	problem	0.647	could	0.679	person
1.288	grateful	0.624	,_and	0.676	read
1.286	dirigibles	0.599	family	0.666	children
1.234	books	0.599	less_time	0.661	offensive_.
1.216	her_new	0.579	spend	0.659	those
...	...	...	...	...	...
1.068	urban_areas	0.343	benefit_our_society	0.480	should_be_able
1.007	airships	0.341	believe_that_computers	0.475	able_to

Table 5: Highest weighted lexical features (i.e. unigrams, bigrams, or trigrams) and their weights in both shared and task-specific representations of the **All-MTL-cTAP** model (associated with results in Table 4) for the two example tasks referred to in Table 1.

both task-specific and shared-representations in a theoretically sound manner (i.e. without making any speculative assumptions about the relative orderings of essays that were graded on different scales using different marking criteria), and is general enough to be used in many situations.

Ultimately these complementary techniques (multi-task learning and constrained pairwise preference-ranking) allow essay scoring data from *any* source to be included during training. As shown in Section 7.2, our approach is robust to increases in the number of tasks, meaning that one can freely add extra data when available and expect the approach to use this data appropriately. This constrained multi-task preference-ranking approach is likely to be useful for many applications of multi-task learning, when the gold-scores across tasks are not directly comparable.

Future work will aim to study different dimensions of the prompt (e.g. genre, topic) using multi-task learning at a finer level. We also aim to further study the characteristics of the multi-task model in order to determine which features transfer well across tasks. Another avenue of potential research is to use multi-task learning to predict scores for different aspects of text quality (e.g. coherence, grammaticality, topicality).

Acknowledgements

We would like to thank Cambridge English Language Assessment for supporting this research, and the anonymous reviewers for their useful feedback. We would also like to thank Ekaterina Kochmar, Helen Yannakoudakis, Marek Rei, and Tamara Polajnar for feedback on early drafts of this paper.

References

Øistein E Andersen, Helen Yannakoudakis, Fiona Barker, and Tim Parish. 2013. Developing and testing a self-assessment and tutoring system. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, BEA*, pages 32–41.

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3).

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 77–80, Sydney, Australia, July. Association for Computational Linguistics.

Ted Briscoe, Ben Medlock, and Øistein Andersen. 2010. Automated assessment of esol free text examinations. Technical Report 790, The Computer Lab, University of Cambridge, February.

Olivier Chapelle and S Sathiya Keerthi. 2010. Efficient algorithms for ranking with svms. *Information Retrieval*, 13(3):201–215.

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*, pages 1741–1752, Seattle, Washington, USA, October. Association for Computational Linguistics.

Hao Cheng, Hao Fang, and Mari Ostendorf. 2015. Open-domain name error detection using a multi-task RNN. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 737–746.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner english: The nus corpus of learner english. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia, June. Association for Computational Linguistics.

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic, June. Association for Computational Linguistics.

Kevin Duh, Katsuhito Sudoh, Hajime Tsukada, Hideki Isozaki, and Masaaki Nagata, 2010. *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, chapter N-Best Reranking by Multitask Learning, pages 375–383. Association for Computational Linguistics.

Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2014, Baltimore, Maryland, USA, June 26-27, 2014*, pages 15–24.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54.

Michael Heilman and Nitin Madnani. 2013. Ets: domain adaptation and stacking for short answer scoring. In *Proceedings of the 2nd joint conference on lexical and computational semantics*, volume 2, pages 275–279.

Derrick Higgins, Jill Burstein, Daniel Marcu, and Claudia Gentile. 2004. Evaluating multiple aspects of coherence in student essays. In *HLT-NAACL*, pages 185–192.

D. Higgins, J. Burstein, and Y. Attali. 2006. Identifying off-topic student essays without topic-specific training data. *Natural Language Engineering*, 12(2):145–159.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM.

N Van Labeke, D Whitelock, D Field, S Pulman, and JTE Richardson. 2013. Openessayist: extractive summarisation and formative assessment of free-text essays. In *Proceedings of the 1st International Workshop on Discourse-Centric Learning Analytics*, Leuven, Belgium, April.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Leah S Larkey. 1998. Automatic essay grading using text categorization techniques. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 90–95. ACM.

Courtney Napoles and Chris Callison-Burch. 2015. Automatically scoring freshman writing: A preliminary investigation. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 254–263, Denver, Colorado, June. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction. In *In Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task (CoNLL-2013 Shared Task)*. Association for Computational Linguistics.

Diane Nicholls. 2003. The cambridge learner corpus: Error coding and analysis for lexicography and elt. In *Proceedings of the Corpus Linguistics 2003 conference*, volume 16, pages 572–581.

Ellis B Page. 1966. The imminence of grading essays by computer. *Phi Delta Kappan*, 47:238–243.

Ellis Batten Page. 1994. Computer grading of student prose, using modern concepts and software. *The Journal of experimental education*, 62(2):127–142.

Ellis Batten Page. 2003. Project essay grade: Peg. *Automated essay scoring: A cross-disciplinary perspective*, pages 43–54.

Isaac Persing and Vincent Ng. 2014. Modeling prompt adherence in student essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1534–1543, Baltimore, Maryland, June. ACL.

Isaac Persing and Vincent Ng. 2015. Modeling argument strength in student essays. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 543–552.

Peter Phandi, 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*, pages 431–439, Lisbon, Portugal, September. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2011. Algorithm selection and model adaptation for esl correction tasks. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 924–933, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lawrence M Rudner and Tahung Liang. 2002. Automated essay scoring using bayes' theorem. *The Journal of Technology, Learning and Assessment*, 1(2).

Swapna Somasundaran, Jill Burstein, and Martin Chodorow. 2014. Lexical chaining for measuring discourse coherence quality in test-taker essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 950–961, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

James H Steiger. 1980. Tests for comparing elements of a correlation matrix. *Psychological bulletin*, 87(2):245.

Helen Yannakoudakis and Ted Briscoe. 2012. Modeling coherence in esol learner texts. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 33–43, Montréal, Canada, June. Association for Computational Linguistics.

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: Human Language Technologies*, pages 180–189, Portland, Oregon, USA, June. Association for Computational Linguistics.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 2*, COLING '00, pages 947–953, Stroudsburg, PA, USA. Association for Computational Linguistics.