UNIVERSITY OF CAMBRIDGE

Modelling text meta-properties in automated text scoring for non-native English writing

Meng Zhang

Churchill College

This dissertation is submitted in May, 2018 for the degree of Doctor of Philosophy

Abstract

Automated text scoring (ATS) is the task of automatically scoring a text based on some given grading criteria. This thesis focuses on ATS in the context of free-text writing exams aimed at learners of English as a foreign language (EFL). The benefit of an ATS system is primarily to provide instant and consistent feedback to language learners, and service reliability also forms a crucial part of an ATS system. Based on previous work, we investigated only partially explored meta-properties in text and integrated them into a machine learning based ATS model across multiple datasets:

- In most previous work, the proposed models implicitly assume that texts produced by learners in an exam are written independently. However, this is not true for the exams where learners are required to compose multiple texts. We hence explicitly instructed our model which texts are written by the same learner, which boosts model performance in most cases.

- We used three intra-exam properties within the same exam including prompt, genre and task as a starting point, and we showed that explicitly modelling these properties via frustratingly easy domain adaptation (FEDA) (Daume III, 2007) can positively affect model performance in some cases. Furthermore, modelling multiple intra-exam properties together is better than modelling any single property individually or no property in four out of five test sets.

- We studied how to utilise and combine learners' responses from multiple writing exams. We also proposed a new variant of the transfer-learning ATS model which mitigates the drawbacks of previous work. This variant first builds a ranking model across multiple datasets via FEDA, and the ranking score of each text predicted by the ranking model is used as an extra feature in the baseline model. This variants gives improvement compared to a baseline model on the development sets in terms of root-mean-square error. Furthermore, the transfer-learning model utilising multiple datasets tuned on each development set is always better than the baseline model on the corresponding test set.

- We found that different datasets favour different meta properties. We therefore combined all the models looking at different meta properties together using

ensemble learning. Compared to the baseline model, the combined model has a statistically significant improvement on all the test sets in terms of root-mean-square error based on a permutation test.

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. This dissertation does not exceed the prescribed limit of 60 000 words.

Meng Zhang
May, 2018

Acknowledgements

This thesis is supported by the institute for Automated Language Teaching and Assessment (ALTA) funded by Cambridge Assessment. I want to acknowledge Cambridge Assessment for the generosity in funding and providing the research data for this thesis. I also want to thank Churchill College and the computer science department for their financial support to conference attendance.

I would like to express my gratitude to my supervisor Ted Briscoe for providing this opportunity to conduct my PhD research in Cambridge. Thanks for your supervision and support for my research.

Next, I would like to appreciate my examiners Paula Buttery and Andreas Vlachos. The quality of this thesis got improved a lot because of the valuable discussion with you during my viva.

I then would like to thank Meichen Lu, Christopher Bryant, Mariano Felice, Ronan Cummins, Yimai Fang, Helen Yannakoudakis and Øistein Andersen for the constructive feedback at different stages. Thank you for helping me make this thesis better.

Finally, I would not have finished this journey without the support from my family and all my friends mentioned and not mentioned in this thesis. You are my heroes in my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The English language is one of the most commonly spoken languages in the world. Based on a report published by the British Council[1], there are about 1.75 billion people who can speak English at a useful level. It is also estimated by the report that the number of English speakers will rise to two billion in 2020. English is also the most widely learned foreign language chosen by the people all around the world. For example, about 97.3% pupils studied English as a foreign language in the European Union (Eurostat, 2017). Besides that, the popularity of learning English is also visible in the number of non-native English speakers taking different English exams. Based on another report from the British Council[2], more than two million International English Language Testing System (IELTS) tests were taken in 2012-2013. The existence of various English exam products provides a useful and fair way for learners to measure their English skills accurately. It also offers a well-accepted standard to help schools, companies and immigration bureaux to quantitatively judge whether the foreign candidates and applicants who are non-native English speakers meet the compulsory language requirements they set up.

To thoroughly and objectively evaluate a learner's English skills from multiple dimensions, some educational organisations like Cambridge Assessment and Educational Testing Service (ETS) propose different types of questions and practices in their English exam products. These questions and practices include not only multiple-choice, true-or-false and fill-in-the-blank style questions, but also more advanced questions such as free-text writing. This variety helps learners identify what they are good at and what their weaknesses are. It also gives them opportunities to practise their English in different ways.

Although the desire to learn English might be high, the hurdle to mastering it is not negligible. One existing barrier is that for some areas, there are not enough human examiners who can objectively evaluate learners' writing quality. While it is relatively

[1] https://www.britishcouncil.org/sites/default/files/english-effect-report-v2.pdf

[2] https://www.britishcouncil.org/organisation/press/record-two-million-ielts-tests

easy to judge the correctness of a multiple-choice question or a true-or-false question if we know what the gold standard is, it requires much more skill to appropriately assess the quality of free-text writing. The lack of professional and qualified examiners makes it hard for learners to get accurate feedback on the quality of their English writing in a timely fashion. Consequently, it is hoped that an automated text scoring (ATS) system can possibly act as a kind of examiner to mitigate this problem, which offers an assistance to both learners and educators.

The main goal of ATS is to automatically evaluate the quality of a person's writing in terms of specific grading criteria. The earliest research into ATS dates back to the 1960s (Page, 1968). It is also hoped that automated assessment will improve consistency and provide assistance in learning and teaching. The application of this technique not only frees educators from the burden of manually marking piles of texts, but also provides learners with instant feedback for iterative writing improvement. The ATS technique is often used as the second marker in high-stakes exams such as e-rater in the Graduate Management Admission Test (GMAT) (Burstein et al., 1998), the Test of English as a Foreign Language (TOEFL) (Chodorow and Burstein, 2004) and the Graduate Record Examinations (GRE) (Ramineni et al., 2012). ATS also acts as the only grader in low-stakes exams and online tutoring applications such as Cambridge English Write & Improve (Andersen et al., 2013) and Turnitin[3].

1.1 Meta Properties

The performance of an ATS system directly determines the success of the automated assessment application. Previous work has approached ATS from different aspects including simple features such as word counts (Page, 1968), semantic similarity (Foltz et al., 1999), prompt relevance (Higgins et al., 2006; Briscoe et al., 2010; Cummins et al., 2016a), syntactic complexity (Yannakoudakis et al., 2011), coherence (Higgins and Burstein, 2007; Yannakoudakis and Briscoe, 2012), discourse relations (Lin et al., 2011; Feng et al., 2014; Song et al., 2017) and argument structure (Klebanov et al., 2016; Wachsmuth et al., 2017). One aspect not fully studied is whether and how the meta properties of a learner's text affect ATS. In this section, we describe how we classify the meta properties of a text into different categories for the research of this thesis.

When an educational organisation designs an exam product, they do so with specific goals and philosophy in mind. For example, some exams are designed to gauge which proficiency level a learner has mastered, while others aim to test whether a learner has fulfilled the requirements for a specific level. In order to benchmark learners' English skills, the Common European Framework of Reference for Languages (CEFR) provides

[3] www.turnitin.com

a common basis to assess language skills in European countries (Council of Europe, 2001). CEFR provides a range of proficiency levels to measure at what stage learners have mastered a specific language. The proposed proficiency is divided into six levels:

- A. Basic User:
  - A1 (Breakthrough)
  - A2 (Waystage)
- B. Independent User:
  - B1 (Threshold)
  - B2 (Vantage)
- C. Proficient User:
  - C1 (Effective Operational Proficiency)
  - C2 (Mastery)

Based on this summary, the IELTS exam is designed to examine which CEFR level learners have achieved. In comparison, some exams like the Cambridge English First (FCE) and Advanced (CAE) exams examine whether participants have mastered a specific CEFR level (B2/C1) or not. Once the ultimate goals and philosophy of an exam product are determined, all the free-text writing questions within the same exam should share the same broad level goals and values on meta properties such as target proficiency level. We define these meta properties as **the intra-exam properties**.

After the goals of the whole exam are decided, the examiners need to design the questions for the exam, and each question has its own focal points. For example, some questions require learners to write an essay, while others ask learners to describe a figure or a picture. Herein, each question also has its own goals, and so we define the meta properties that distinguish a question from other questions within the same exam as **the inter-exam properties**.

When the exam organisation publishes their exam products, learners can then register to take the exams they want. Other meta properties that distinguish different writing could be but not limited to participant gender, native language and age. We define the meta properties which are independent of an exam but characterised by the external properties of participants or the environment where participants take the exam as **the exam-independent properties**. In contrast, we put the intra-exam and inter-exam properties under a broader term together as **the exam-dependent properties**. The values of the exam-dependent properties of a text rely solely on the exam product, while the exam-independent properties are independent of the exam, the designers of the exam and the questions in the exam. For example, two responses to the same prompt in the same exam can have different values for these exam-independent properties, but they must have the same value on any exam-dependent property. The relations between these meta properties are shown in Figure 1.1.

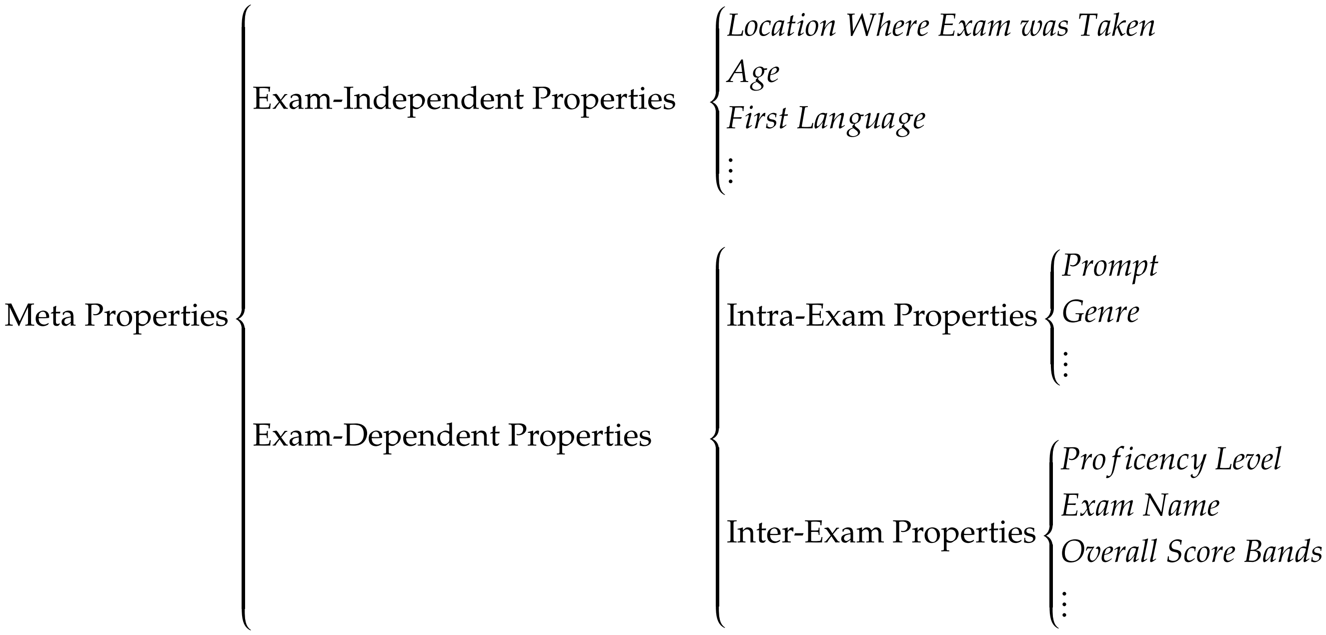


Figure 1.1: Meta properties of a learner's text

1.2 Research Questions

Obviously, some meta properties we list here affect and determine the grading criterion of a text, and explicitly differentiating meta-proprieties of different texts in an ATS system might be beneficial in improving system performance. It is always crucial to provide learners with accurate feedback regarding their writing quality. This thesis focuses on meta properties that have not been thoroughly examined in previous work. Specifically, we try to determine how text meta properties can be used in an ATS system. We also intend to investigate whether adding and explicitly modelling these meta properties can improve ATS system performance or not.

1.3 Thesis Structure and Contributions

The remainder of this thesis is structured:

- Chapter 2 reviews previous work in ATS.

- Chapter 3 describes how we prepare the datasets and baseline model for this thesis. We prepared six different datasets and extended the model proposed by Yannakoudakis et al. (2011). Based on the experimental results in this chapter, we chose the support vector regression model as the baseline model in the following chapters.

Chapters 4 to 6 take a closer look at three types of meta properties in ATS. The contents and the major contributions of each chapter are listed as follows:

- Chapter 4 focuses on the authorship knowledge of a text to study the exam-independent property in this chapter. In some English writing exams, learners are required to write texts to multiple free-text writing tasks. Previous work has implicitly assumed that all texts are composed independently and so ignores the fact that some texts are actually composed by the same learner in a single exam session. We define this information as the authorship knowledge of a text. In this chapter, we proposed two approaches to explicitly pass this knowledge to the baseline model by means of feature fusion and score fusion. The benefit of doing this is to add more connections among the training data so that the model can better understand a learner's writing skills. We experimentally demonstrated that model performance improves in most cases, and that this improvement is irrespective of human examiners' bias of knowing the authorship of each text even if this bias might exist.

- Chapter 5 looks at the intra-exam properties of a text. In this chapter, we examined three properties including prompt, genre and task as a starting point. We experimentally demonstrated that modelling these properties via frustratingly easy domain adaptation (FEDA) (Daume III, 2007) improves our grader performance on some datasets.

- Chapter 6 investigates the modelling of the inter-exam properties of a text. Particularly, we distinguished which exam each dataset comes from and studied whether utilising multiple datasets is better than building a grader from only a single dataset.

  In this chapter, we proposed a variant based on the model from Cummins et al. (2016b). They built a pairwise ranking model based on FEDA across multiple datasets and predicted a *ranking score* for each text, and they then used a linear regression model to map the outputs of the ranking model back to the original scores for each exam. We found that their model performed worse than the baseline model in terms of root-mean-square error on some development sets. To overcome this problem, we used the predicted ranking score of each text as an extra feature and fed it back to the baseline model in Chapter 3. Compared to the other models evaluated in this thesis, this variant gets the best performance on four out of six development sets in terms of the root-mean-squared error.

Finally, the best-performing transfer-learning model utilising multiple datasets tuned on each development set is always better than the baseline model in Chapter 3 on the corresponding test set.

As we found that different datasets benefit from different meta properties proposed in Chapter 4 to Chapter 6, in Chapter 7, we jointly modelled all the meta properties together via ensemble learning. Compared to the baseline model in Chapter 3, the combined model has a statistically significant improvement on all the test sets in terms of root-mean-square error based on the permutation test.

Finally, Chapter 8 draws the conclusions and identifies possible future work for this thesis.

Chapter 2

Background

This chapter summarises the datasets and techniques that have been used in ATS. A more comprehensive survey can be found in Shermis and Burstein (2003); Valenti et al. (2003); Shermis and Burstein (2013); Burrows et al. (2015).

2.1 Learner Corpora

It is difficult to conduct research into ATS without a suitable learner corpus. A good learner corpus is required to provide an opportunity for us to conduct quantitative analyses and test assumptions about ATS. The quality of this corpus directly determines the quality of our work/research/models. In this section, we introduce several learner corpora that have been used for various purposes in previous work. The corpora not covered in this section can be found in the surveys written by Pravec (2002); Granger (2004); Nesselhauf (2004). A comprehensive list of learner corpora is available on the website of the Univeristé catholique de Louvain.[1]

Many datasets have been created by different researchers and organisations. Building a learner corpus has substantially benefited various natural language processing tasks including:

- first language identification (Gebre et al., 2013; Tetreault et al., 2013, 2012; Brooke and Hirst, 2012; Kochmar, 2011; Koppel et al., 2005);
- grammatical error detection and correction (Ji et al., 2017; Yuan, 2017; Chollampatt et al., 2016; Herbelot and Kochmar, 2016; Rei et al., 2016; Yuan et al., 2016; Susanto et al., 2015; Kochmar, 2016; Andersen, 2011; De Felice and Pulman, 2009, 2008);
- grammatical error correction evaluation (Felice et al., 2016; Bryant and Felice, 2016; Napoles et al., 2016; Felice and Briscoe, 2015; Dahlmeier and Ng, 2012);

[1] https://uclouvain.be/en/research-institutes/ilc/cecl/learner-corpora-around-the-world.html

- artificial grammatical error generation (Felice, 2016; Felice and Yuan, 2014; Foster and Andersen, 2009);
- part-of-speech (POS) tagging and parsing of texts written by learners of English as a foreign language (EFL) (Berzak et al., 2016);
- analysing the similarity between learners' first languages via their second language writing patterns (Berzak et al., 2015, 2014);
- ATS, which is the topic of this dissertation.

These corpora can be characterised by several other criteria. For example:

- the medium in which the learners' responses are collected (hand-written or computer-typed);
- the learners' first languages (English, Chinese, German);
- the text type (narrative, discussion, exams, assignments);
- the learners' proficiency levels (beginning, intermediate or advanced);
- whether the corpus is publicly available or has limited access.

In this thesis, we are primarily interested in learners' abilities in written English. One way to classify these corpora is based on whether the learners' first language is English or not.

2.1.1 English Native Speaker Corpora

One of the most well-known publicly-available corpora for English native speakers is the Automated Student Assessment Prize (ASAP) dataset.[2] The William and Flora Hewlett Foundation (Hewlett) organised a shared task on ATS in 2013. In this shared task, Hewlett gathered about twenty thousand texts from eight different prompts. The essays were written by students in Grade 7 to Grade 10 in the United States, and each essay is between 150 to 550 words, depending upon the requirement of each prompt. The dataset has been used in previous work in ATS (Chen and He, 2013; Phandi et al., 2015; Alikaniotis et al., 2016; Cummins et al., 2016b; Taghipour and Ng, 2016; Dong and Zhang, 2016; Dong et al., 2017; Tay et al., 2018; Cozma et al., 2018).

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

2.1.2 EFL Corpora

In this dissertation, we are more interested in EFL learners corpora, as human examiners who can objectively assess learners' writing are hard to access for EFL learners in some areas. That said, the research outcome from this dissertation should also be of benefit to ATS for native speakers even if we only use the EFL learners corpora to conduct our research. In this section, we introduce several EFL corpora and identify whether they are suitable for our research into ATS in this dissertation.

The International Corpus of Learner English (ICLE) (Granger et al., 2002) is one of the biggest and earliest corpora built for essays written by EFL learners. The first and second versions were released in 2002 and 2009, respectively. The second version contains argumentative essays and literature examination papers with 3.7 million words written by intermediate and advanced learners. There are sixteen different first language backgrounds for these learners including Bulgarian, Chinese, Czech, Dutch, Finnish, French, German, Italian, Japanese, Norwegian, Polish, Russian, Spanish, Swedish, Tswana and Turkish.[3] Each essay contains 500-1000 words. Data collection involved the collaboration of a group of participating universities. In this corpus, only prompts were controlled by Granger et al., and other factors were left to the participating universities to decide. For example, different students wrote texts in different environments and it is unclear whether they used dictionaries and grammar books or not. Although the second version of the corpus included meta-data such as gender, first, second or even third language, it did not contain any scores indicating the quality of each essay. Persing et al. consequently marked a sample of essays from this corpus based on organisation (Persing et al., 2010), thesis clarity (Persing and Ng, 2013), prompt adherence (Persing and Ng, 2014) and argument strength (Persing and Ng, 2015).

The National University of Singapore Corpus of Learner English (NUCLE) was created by Dahlmeier et al. (2013). It contains 1,400 essays with one million words written by students from the National University of Singapore on different topics. All the essays were annotated with error tags and corrections by well-trained English instructors, although the essays themselves are not marked with any grading criterion. Consequently, NUCLE is not directly suitable for research in ATS. Many corpora share the same problem that they are not marked based on any grading criteria such as the Langman Learner's Corpus (LLC) (Gillard and Gadsby, 1998) and the Uppsala Student English (USE) corpus (Axelsson, 2000).

The Cambridge Learner Corpus (CLC) has been developed through a collaboration between Cambridge University Press (CUP) and Cambridge Assessment English since 1993 (Nicholls, 2003). The CLC contains 52.5 million words (16 million in 2003) written by non-native English learners in different types of exams. Each text was marked with

[3] https://uclouvain.be/en/research-institutes/ilc/cecl/icle.html

a score based on the grading criterion of the corresponding exam. The 2003 version has 86 different first languages, and the 350,000-word subset of the corpus is further annotated with 80 different error codes. Although access to the full CLC is restricted by CUP and its collaborators, a subset is publicly available. Specifically, the FCE public (FCE-PUB) dataset was released by Yannakoudakis et al. (2011). This dataset contains the writing sections from 1,141 exam papers answered by 1,141 learners in the year 2000 and 103 exam papers written by 103 learners in 2001.

TOEFL11 (Blanchard et al., 2013) are the essays collected from the Test of English as a Foreign Language (TOEFL) organised by the Educational Testing Service. This dataset was originally collected and prepared for the purpose of native language identification, but it can also be used in other research projects in education. It contains 12,100 essays with 11 different first languages, evenly sampled, including Arabic, Chinese, French, German, Hindi, Italian, Japanese, Korean, Spanish, Telugu and Turkish. Each essay in this dataset was marked with one out of three different levels (low/medium/high). Blanchard et al. pointed out one difference between the FCE-PUB public dataset and TOEFL11 is that the FCE-PUB dataset has a broader range of prompts available.

2.1.2.1 English Exams from Cambridge Assessment

In this dissertation, we use learners' data collected from the English exam products designed by Cambridge Assessment. There are several advantages of using the Cambridge Assessment exams to conduct our research.

An advantage of the English exam products made by Cambridge Assessment is that the marking scheme and quality of each Cambridge English exam is strictly controlled (Ffrench et al., 2012). In particular, all examiners need to complete a training class and pass a qualification assessment in order to be qualified to assess learners' writings. Next, during the marking procedure, the performance of each examiner is monitored and regularly reviewed by senior examiners to ensure that the scores assigned by examiners accurately reflect the writing ability of each learner. This monitoring procedure also provides instant feedback to examiners concerning their scoring performance. which ensures the quality of the dataset we use in this thesis.

In fact, such strict control is not only applied to the marking procedure but also to the environment in which learners compose and finish their writing. For instance, unlike in ICLE, the academic writing and general writing tasks in IELTS must be strictly completed within one hour.[4] This guarantees that learners taking exams in different locations are all overseen and examined under the same conditions.

Another benefit of Cambridge English exams is that participating learners have different first languages, and they come from a variety of countries and cultures.

[4] https://www.ielts.org/en-us/about-the-test/test-format-in-detail

The CLC, for example, has collected responses from learners with 86 different first languages (Nicholls, 2003). This diversity can help us ensure the robustness of the automated grader we build. In other words, the grader is trained to capture the linguistic competency of each learner, regardless of their first language or cultural background. Also, different English exams in the CLC target learners of different CEFR levels, so we can investigate how the target proficiency level of an exam affects automated assessment. In contrast to the ICLE and TOEFL11 corpora, there are also different types/genres of text in each exam, and, similarly, this can be used to evaluate the robustness of the automated grader.

Our work is an extension of the previous work done by Briscoe et al. (2010); Yannakoudakis et al. (2011); Yannakoudakis and Briscoe (2012) (to be described in Section 2.4), who evaluated their approach on the same subset of the CLC. A common problem in ATS is that different approaches are not evaluated on the same dataset. For example, although some ATS systems (Page and Petersen, 1995; Cozma et al., 2018) have achieved reasonably good performance and even outperformed human annotation agreement on some datasets it is actually hard to interpret which system is better, because they all evaluated on their own private datasets. Hence, to benchmark with previous work, we hence test our hypotheses on the same CLC subset released by Yannakoudakis et al. (2011).

Finally, the CLC is the biggest dataset our research group can access. Due to licensing issues, many corpora such as the Hong Kong University of Science and Technology (HKUST) corpus (Milton and Chowdhury, 1994) are unobtainable. In contrast, our research partner Cambridge Assessment is willing to share their dataset to test our assumptions and methods in this thesis.

2.1.2.2 Product Categories

The products of Cambridge English exams can be classified into the following categories:

- The main suite consists of a series of exams including Key English Test (KET), Preliminary English Test (PET), First Certificate of English (FCE), Certificate of Advanced English (CAE) and Certificate of Proficiency in English (CPE). These exams target learners with different CEFR language proficiency levels;
- The business suite (BEC) focuses on business English (e.g. business preliminary, business vantage and business higher) with different CEFR language proficiency levels;
- The universal exams without a target proficiency level include the International English Language Teaching System (IELTS) and the Business Language Testing Service (BULATS).

All the Cambridge English exams have writing sections to examine and gauge learners' writing abilities. The writing sections in the main suite and business suite are designed to target specific proficiency levels. In contrast, the universal exams do not target a specific proficiency level. In this thesis, we investigate the exams in the main suite and IELTS. We will describe the exams and their corresponding datasets in more detail in Chapter 3.

2.2 Machine Learning

Most previous work into ATS relies on machine learning to model the relationship between texts and scores (Page, 1968; Larkey, 1998; Yigal and Burstein, 2006; Yannakoudakis, 2013; Zesch et al., 2015; Dong and Zhang, 2016; Taghipour and Ng, 2016; Tay et al., 2018). In this section, we introduce several machine learning models used in ATS. More details about different machine learning techniques can be found in the textbooks and surveys of Mitchell (1997); Bishop (2006); Murphy (2012); Goodfellow et al. (2016). The aim of machine learning is to implement a system capable of learning from an experience related to a task to improve the system performance in this task based on a performance measure (Mitchell, 1997).

There are several tasks in machine learning, one of which is *supervised learning*. In supervised learning, the system learns a function f that predicts output $y_i \in \mathbb{Y}$ given input $x_i \in \mathbb{X}$ without explicit human instructions. In automated assessment, the output could be a scalar y_i, which is the score of the text marked by a human examiner, or a vector $\mathbf{y}_i$, which represents different aspects of the quality of a text. In this thesis, we assume each output is a scalar y_i. The input x_i is a text written by a learner. This learning process requires an annotated dataset, a machine learning model and an appropriate optimisation method to learn the relationship between the input and output space. In automated assessment, the annotated dataset consists of a collection of texts with scores marked by human examiners; we define this dataset as the **training set** $\{x\}_{\text{train}}$ in supervised learning. The procedure for the system to learn this relationship is called **training**. The machine learning model can only read data in a specific format, and the process that maps each text to a model-readable format is defined as **feature extraction**. The format of the model-readable data of x_i could hence be a vector $\mathbf{x}$ with size D:

$$\mathbf{x}_i =< x_{i,1}, x_{i,2}, \ldots, x_{i,D} >$$

It could also be a sequence of vectors or other possible forms depending on how the model is designed. During training, we want the predictions $\hat{y}_i$ made by the model on the input x_i to be as close as possible to the gold score y_i. We therefore need some

performance measures P to quantitatively describe this closeness, and the model should be optimised on a pre-defined target function which reflects the performance measure we are interested in. In other words, P should ideally get better when we are optimising the pre-defined target function.

After we train a model, we need to ensure the model performs well on future unseen data. To evaluate future performance, we reserve an annotated dataset and evaluate the trained model on this held-out dataset. This procedure is called **testing**, and the unseen annotated dataset $\{x\}_{\text{test}}$ used in testing is the **test set**. The format of the training and test sets should be the same, and they should both have been marked and annotated by human examiners.

In order to optimise on the performance measure P, we need to define some configurational parameters before training. Compared to the parameters of the model directly learned during the training process, these configurational parameters cannot be directly learned from the training process. These configurational parameters are called the **hyper-parameters** of the model. The hyper-parameters of a model could be the parameters controlling the complexity of the parameters learned by the model including the regularisation term, the learning rate of the optimisation method, or the time when we stop training the model. During training, when deciding appropriate values for hyper-parameters, we can tune the model on a **development set**, which is another annotated dataset following the same annotation procedure as the training and test sets. We use the optimal hyper-parameters on the development set in the trained model we evaluate on the test set. This hyper-parameters tuning process is called **validation**.

In contrast to supervised learning, *unsupervised learning* does not involve any annotated dataset but identifies the pattern and structure of some unannotated data that might be useful for some downstream tasks. At the intersection of supervised learning and unsupervised learning is *semi-supervised learning*. Semi-supervised utilises both labelled and unlabelled data instances to learn the relation between the input space $\mathbb{X}$ and the output space $\mathbb{Y}$. Only limited work (Chen et al., 2010) in automated assessment studied unsupervised learning.[5] Most previous work in this field relies on supervised learning (Page, 1968; Larkey, 1998; Briscoe et al., 2010; Yannakoudakis, 2013). Our work in the following chapters is also based on supervised learning because the patterns learned from unsupervised learning might not capture the patterns to predict the scores we are interested in. For this reason, we focus on supervised learning in this thesis.

[5] Although Chen et al. (2010) did not explicitly use essay scores in training their model; their work still needs some supervised signals, such as knowing the historical score distribution so that they knew the number of unique terms in each essay has a strong correlation with the essay scores in their dataset.

2.3 Supervised Learning

Generally speaking, three different board supervised learning methods that have previously been applied to automated assessment include regression, classification and ranking. In regression, the model predicts a score on a continuous scale, while in classification, it predicts a label from a finite set. For example, it would be a regression task to predict the amount of rainfall on a particular day, but a classification task to predict whether it will rain or not. This latter case is a binary-classification task. In contrast, a ranking model aims to determine the order of a set of inputs. For example, we can rank essays from best to worst based on overall quality.

In this section, we will describe several machine learning models used in automated assessment, which cover the necessary knowledge required to understand the techniques relevant to this dissertation.

2.3.1 Linear Regression

We start with the simplest regression model: linear regression. Linear regression is a regression technique. It assumes that given a data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ of N instances, the input space $\mathbb{X}$ and the output space $\mathbb{Y}$ form a linear relationship. The prediction function f for a given data instance $\mathbf{x}_i \in \mathbb{R}^D$ is defined as follows:

$$f(\mathbf{x}_i; \mathbf{w}, b) = \mathbf{w}^T \mathbf{x}_i + b = \hat{y}_i \tag{2.1}$$

$\hat{y}_i$ is the value predicted by f for $\mathbf{x}_i$; $\mathbf{w} \in \mathbb{R}^D$ and $b \in \mathbb{R}$ are the model parameters to be learned in training. During training, the target function we would like to optimise can be described in Equation 2.2:

$$\min_{\mathbf{w},b} Cost(\mathbf{w}, b) = Error(\hat{\mathbf{y}}, \mathbf{y}) + \lambda Reg(\mathbf{w}, b) \tag{2.2}$$

The first term $Error(\hat{\mathbf{y}}, \mathbf{y})$ in this equation means that we want to minimise the error between the predictions $\hat{\mathbf{y}}$ and gold labels $\mathbf{y}$. Some common metrics quantifying $Error(\hat{\mathbf{y}}, \mathbf{y})$ include the mean-squared error (L2 error):

$$Error_{L2}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{2.3}$$

and the mean-absolute error (L1 error):

$$Error_{L1}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i| \tag{2.4}$$

The second term in the target function Equation 2.2 is the regularisation term that simplifies the model complexity. After training, we expect the model not only to correctly mark texts in the training set, but also to accurately predict scores for unseen data instances. In other words, rather than merely memorising and overfitting the seen data, we want the model to generalise well on the unseen data. Adding a regularisation term $\lambda Reg(\mathbf{w}, b)$ into the target function can hence avoid the model overfitting to the noise in the training set. λ is the hyper-parameter to balance the contribution between the error term and the regularisation term in the target function. Commonly used regularisation terms include the L2 penalty:

$$Reg_{L2}(\mathbf{w}, b) = ||\mathbf{w}||_2^2 + b^2 = \sum_{d=1}^{D} w_d^2 + b^2 \tag{2.5}$$

and the L1 penalty:

$$Reg_{L1}(\mathbf{w}, b) = ||\mathbf{w}||_1 + |b| = \sum_{d=1}^{D} |w_d| + |b| \tag{2.6}$$

The target function Equation 2.2 can be optimised by different approaches, the most well-known of which is gradient descent. In this approach, at each iteration, we calculate the partial derivatives of the parameters $\mathbf{w}$ and b over the target function $Cost(\mathbf{w}, b)$ with $\frac{\partial Cost}{\partial \mathbf{w}}$ and $\frac{\partial Cost}{\partial b}$. We then update each parameter by adding its negative partial derivative:

$$\begin{aligned} \mathbf{w} &:= \mathbf{w} - \eta \frac{\partial Cost(\mathbf{w}, b)}{\partial \mathbf{w}} \\ b &:= b - \eta \frac{\partial Cost(\mathbf{w}, b)}{\partial b} \end{aligned} \tag{2.7}$$

η is the learning rate hyper-parameter to be configured before we start to train the model. We repeat this procedure until $\mathbf{w}$ and b converge. It has been proven that gradient descent always converges to the global minimum if the target function is strictly convex and the learning rate η is appropriately chosen (Boyd and Vandenberghe, 2004). More advanced optimisation methods are also available, but they are not the focus of this dissertation so we refer the reader to Goodfellow et al. (2016); Boyd and Vandenberghe (2004) for more information.

2.3.2 Support Vector Machines

The support vector machine (SVM) is a popular machine learning technique proposed by Cortes and Vapnik (1995) used for classification. It has shown good performance in a wide variety of natural language processing tasks (Joachims, 1998b). The explanation in

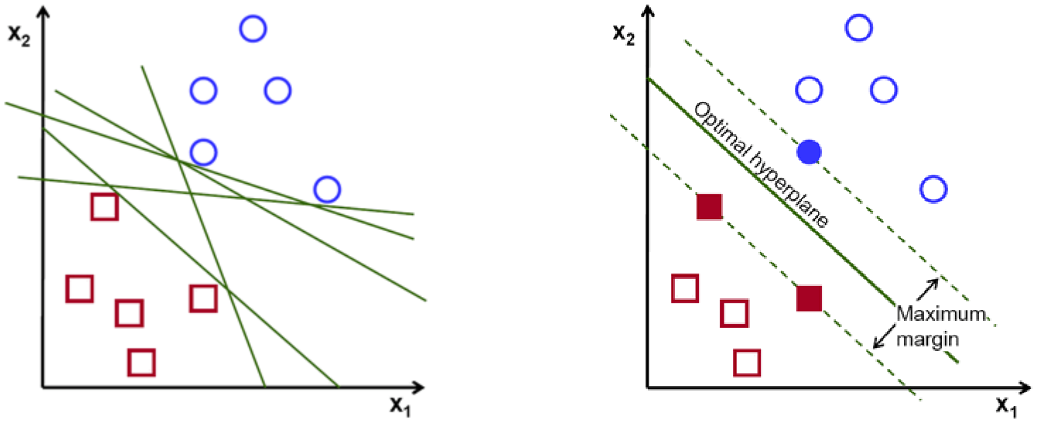


Figure 2.1: SVM hyperplanes that separate blue and red points. Left: some valid hyperplanes separating the given points; Right: the optimal-hyperplane found by the SVM which maximises the distance (margin) between the two dashed hyperplanes bounding the blue and red points, respectively.

this section can be found in the notes written by Ng[6].

An SVM model is a maximum-margin classifier. In order to explain the intuition behind maximum margin, let us start with binary classification. In Figure 2.1[7], given N points $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, we are asked to find a (green) boundary to separate these N positive and blue ($y = 1$) and negative and red ($y = -1$) points. This hyperplane can be described by the set of points $\mathbf{x}$ satisfying the following equation:

$$\mathbf{w}^T\mathbf{x} + b = 0 \tag{2.8}$$

In the SVM, it is assumed that the ideal boundary we are looking for should maximally separate the positive and negative data instances. If the data points are linearly separable by the ideal boundary, there should exist another two hyperplanes (the two green dashed lines in Figure 2.1) in parallel to the ideal boundary. Each dashed hyperplane bounds all the points from the same category. The distance between these two dashed hyperplanes is called the margin in the SVM, and the ideal hyperplane lies halfway between these two dashed hyperplanes. The data points (the red and blue **solid** points) lying on the two dashed hyperplanes are the support vectors. The two dashed hyperplanes can respectively be described by the equations

$$\mathbf{w}^T\mathbf{x} + b = -1 \tag{2.9}$$

and

$$\mathbf{w}^T\mathbf{x} + b = +1 \tag{2.10}$$

[6] http://cs229.stanford.edu/notes/cs229-notes3.pdf

[7] Source: http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

In order to maximise the margin between the two hyperplanes $\frac{2}{||\mathbf{w}||}$, we can reformulate this problem as:

$$\begin{gathered}\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 \\ \text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \ \forall i \in 1, \ldots, N\end{gathered} \tag{2.11}$$

This is a quadratic convex optimisation problem that can be solved using existing optimisation methods such as the ellipsoid method (Grötschel et al., 2012, p.64).

So far, we have assumed that the points from negative and positive classes are linearly separable. In order to make the algorithm work in the non-linearly separable case and also more robust to the outliers in data, we add a slack variable ξ_i for each data instance $(\mathbf{x}_i, y_i)$ as:

$$\xi_i = \max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i - b)) \tag{2.12}$$

and we reformulate the minimisation problem in the following way:

$$\begin{gathered}\min_{\mathbf{w},\xi} C\sum_{i=1}^{N} \xi_i + \frac{1}{2}||\mathbf{w}||^2 \\ \text{subject to } \begin{cases} y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \ \forall i \in 1, \ldots, N \\ \xi_i \geq 0, \forall i \in 1, \ldots, N \end{cases}\end{gathered} \tag{2.13}$$

C is the hyper-parameter to balance the degree of error we can tolerate and how much sacrifice we can accept if the margin found by the model becomes smaller. This is similar to the hyper-parameter controlling the regularisation term in linear regression (Equation 2.2).

The optimisation problem we are solving now is defined as the *primal optimisation problem*. In contrast, we can also turn the SVM optimisation into a dual problem. Solving the dual problem can help us know the contribution of each data instance in deciding the final learned hyperplane in the SVM model. For Equation 2.13, we first wrap the target function we are minimising and the constraints together into $L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{a}, \boldsymbol{r})$ (*the generalised Lagarangian*) by the Lagrange multiplier:

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{a}, \boldsymbol{r}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i[y_i(\mathbf{x}_i^T\mathbf{w} + b) - 1 + \xi_i] - \sum_{i=1}^{N} r_i\xi_i \tag{2.14}$$

Here we define:

$$f_0(\mathbf{w}, \boldsymbol{\xi}) \equiv \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N} \xi_i \tag{2.15}$$

$$f_i^{\alpha}(\mathbf{w}, b, \xi_i) \equiv -[y_i(\mathbf{x}_i^T\mathbf{w} + b) - 1 + \xi_i] \tag{2.16}$$

$$f_i^{r}(\xi_i) \equiv -\xi_i \tag{2.17}$$

We represent $\theta_{primal}(\mathbf{w}, b, \boldsymbol{\xi})$ as $\max_{\boldsymbol{\alpha},r:\boldsymbol{\alpha},r\geq\mathbf{0}} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{a}, \boldsymbol{r})$, and $\min_{\mathbf{w},b,\xi} \theta_{primal}(\mathbf{w}, b, \boldsymbol{\xi})$ is equivalent to the primal optimisation problem in Equation 2.13:

$$p^* = \min_{\mathbf{w},b,\xi} \theta_{primal}(\mathbf{w}, b, \boldsymbol{\xi}) = \min_{\mathbf{w},b,\xi} \max_{\alpha,r:\boldsymbol{\alpha},r\geq\mathbf{0}} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{a}, \boldsymbol{r}) \tag{2.18}$$

If we swap the minimum-maximum order in this equation, we get the *dual optimisation problem* $\max_{\boldsymbol{\alpha},r:\boldsymbol{\alpha},r\geq\mathbf{0}} \theta_{dual}(\boldsymbol{\alpha}, \boldsymbol{r})$:

$$d^* = \max_{\alpha,r:\boldsymbol{\alpha},r\geq\mathbf{0}} \theta_{dual}(\boldsymbol{\alpha}, \boldsymbol{r}) \equiv \max_{\alpha,r:\boldsymbol{\alpha},r\geq\mathbf{0}} \min_{\mathbf{w},b,\xi} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{r}) \tag{2.19}$$

The solution to the dual (maximisation) problem d^* is the lower bound to the solution of the primal (minimisation) problem p^*:

$$d^* \leq p^* \tag{2.20}$$

Based on the Slater condition (Slater, 1959), the lower bound provided by the dual problem is exactly the optimal solution for the primal problem, because f_0, f_i^{α} and f_i^{r} are all convex functions. Moreover, the optimal solutions $\mathbf{w}^*$, b^*, $\boldsymbol{\xi}^*$, $\boldsymbol{\alpha}^*$ and $\boldsymbol{r}^*$ should satisfy the Karush-Kuhn-Tucker (KKT) conditions (Karush, 1939; Kuhn and Tucker, 1951):

$$\begin{aligned}
\frac{\partial}{\partial w_i} L(\mathbf{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{\alpha}^*, \boldsymbol{r}^*) &= 0,\ i = 1, \ldots, N \\
\frac{\partial}{\partial \xi_i} L(\mathbf{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{\alpha}^*, \boldsymbol{r}^*) &= 0,\ i = 1, \ldots, N \\
\frac{\partial}{\partial b} L(\mathbf{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{\alpha}^*, \boldsymbol{r}^*) &= 0 \\
a_i^* f_i^{\alpha}(\mathbf{w}^*, b^*, \xi_i^*) &= 0,\ i = 1, \ldots, N \\
r_i^* f_i^{r}(\xi_i^*) &= 0,\ i = 1, \ldots, N \\
f_i^{\alpha}(\mathbf{w}^*, b^*, \xi_i^*) &\leq 0,\ i = 1, \ldots, N \\
f_i^{r}(\xi_i^*) &\leq 0,\ i = 1, \ldots, N \\
\alpha_i^* &\geq 0,\ i = 1, \ldots, N \\
r_i^* &\geq 0,\ i = 1, \ldots, N
\end{aligned} \tag{2.21}$$

Based on the KKT conditions, we can substitute $\mathbf{w}$, b, $\boldsymbol{\xi}$ and $\boldsymbol{r}$ by $\boldsymbol{\alpha}$ and further turn $\max_{\boldsymbol{\alpha},\boldsymbol{r}:\boldsymbol{\alpha},\boldsymbol{r}\geq\mathbf{0}} \theta_{dual}(\boldsymbol{\alpha},\boldsymbol{r})$ into the following form:

$$
\begin{aligned}
&\max_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\
&\text{subject to } \begin{cases} 0 \leq \alpha_i \leq C \; \forall i \in 1, \ldots, N \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \end{cases}
\end{aligned}
\tag{2.22}
$$

where $K(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function that encodes the relationship between $\mathbf{x}_i$ and $\mathbf{x}_j$. More specifically, $K(\mathbf{x}_i, \mathbf{x}_j)$ is the inner product between vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ if K is a linear kernel. We can also use other kernels such as a polynomial or radial basis function kernel to map $\mathbf{x}$ into a non-linear space, which might improve the model's capability of capturing more useful patterns in classification and handling of cases when data are not linearly separable.

By treating the primal problem as a dual problem, it becomes easier and faster to do non-linear transformations on the $\mathbb{X}$ space by applying different kernels including the radial basis function and polynomial kernels. In addition, some optimisation methods such as sequential minimal optimisation (Platt, 1998) and tricks such as heuristic shrinkage (Joachims, 1998a) can accelerate the convergence rate and reduce the memory usage of SVMs. Besides that, by combining the primal and dual optimisation problems together, Hsieh et al. (2008) further speeds up SVMs by the dual coordinate descent method based on the linear kernel.

2.3.3 SVM Pairwise Ranking

An easy approach to solve a ranking problem by SVMs is Ranking SVM (Joachims, 2002). This method treats ranking as a pairwise ranking problem. In this problem, we have N texts, and r is the set of all the pairs of texts x_i and x_j where x_i receives a higher score than x_j has. The model needs to correctly decide which text should get a higher score in each pair of texts. We can apply the binary classification framework used in the SVM to solve the ranking problem based on a linear kernel by:

$$
\begin{aligned}
&\min_{\mathbf{w},\xi} C \sum_{(x_i,x_j)\in r} \xi_{i,j} + \frac{1}{2}||\mathbf{w}||^2 \\
&\text{subject to } \begin{cases} \mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j) \geq 1 - \xi_{i,j} \; \forall (x_i, x_j) \in r \\ \xi_{i,j} \geq 0, \forall (x_i, x_j) \in r \end{cases}
\end{aligned}
\tag{2.23}
$$

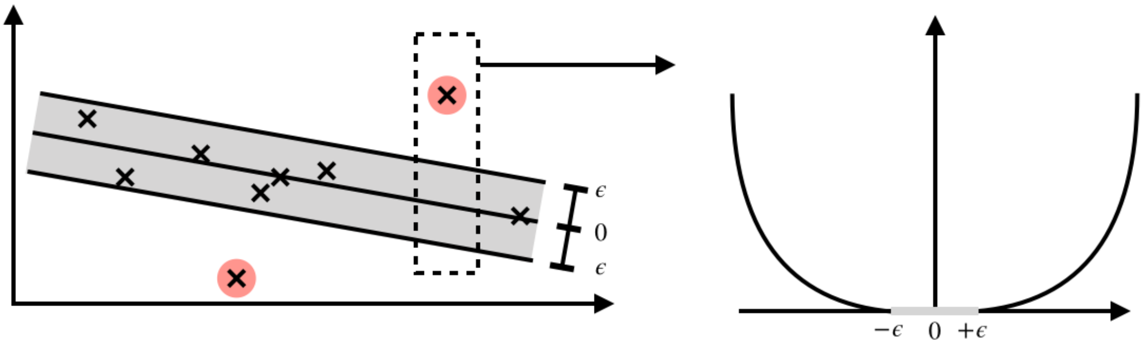


Figure 2.2: Illustration of SVR. Left figure: The black boundaries of the grey region represent the upper and lower boundaries of SVR. The scores of the points in the grey area can be correctly predicted by SVR, and the score errors predicted by the SVR model for the points outside of the grey region are zero (Equation 2.24). Right figure: the error function of L2 loss in SVR.

The number of constraints is twice the size of all the possible pairs r, and we can use the same optimisation strategy used in the SVM to learn the model optimal parameters as $\mathbf{w}^*$.

During prediction, given a dataset $\{x_i\}_{i=1}^N$, we calculate the ranking score $\hat{y}_i^{rank}$ for each data instance x_i by $\mathbf{w}^* \cdot \mathbf{x}_i$, and the rank of $\{x_i\}_{i=1}^N$ is sorted based on $\{\hat{y}_i\}_{i=1}^N$. The ranking problem is also solvable in the dual form in the SVM with different kernels other than the linear kernel.

2.3.4 Support Vector Regression

For the target function of linear regression model described in Equation 2.2, we can see that the regression problem can actually be fitted into the SVM framework in Equation 2.13 as support vector regression (SVR) (Vapnik, 1999; Smola and Schölkopf, 2004), because both equations contain the error and regularisation terms to be minimised. Based on the optimisation problem for the SVM in Equation 2.13, the regression problem based on a linear kernel is formulated by adding the error constraints in both directions with L2 loss by:

$$\begin{aligned} &\min_{\mathbf{w},b,\xi^{upper},\xi^{lower}} C\sum_{i=1}^{N}((\xi_i^{upper})^2 + (\xi_i^{lower})^2) + \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to} \begin{cases} y_i - \mathbf{w}^T\mathbf{x_i} - b \le \epsilon + \xi_i^{upper} \\ \mathbf{w}^T\mathbf{x_i} + b - y_i \le \epsilon + \xi_i^{lower} \qquad \forall i \in 1,\dots,N \\ \xi_i^{upper}, \xi_i^{lower} \ge 0 \end{cases} \end{aligned} \tag{2.24}$$

where C is the cost hyper-parameter, and ϵ is the hyper-parameter controlling the margin of tolerance where no penalty is given to errors. ξ_i^{upper} and ξ_i^{lower} describe how much error the model makes on $\mathbf{x}_i$ in terms of the upper and lower boundaries in the SVR (Figure 2.2). Similarly to the SVM, this regression problem can still be converted to its dual form so that we can find the support vectors of SVR. After we find the optimal parameters $\mathbf{w}^*$ and b^* during training, we can easily do prediction for an incoming instance x_i as $\hat{y}_i = \mathbf{w}^* \cdot \mathbf{x}_i + b^*$.

2.4 Related Work in ATS

Many researchers have studied different aspects of ATS Some work focus on scoring on different types of texts including short text (Leacock and Chodorow, 2003b; Sukkarieh et al., 2003; Pulman and Sukkarieh, 2005; Mohler et al., 2011; Tack et al., 2017), narration (Somasundaran et al., 2015, 2018), academic writing (Yang et al., 2018) and essay scoring. Automated essay scoring could be further split into non-English (Ishioka and Kameda, 2006; Östling et al., 2013; Horbach et al., 2017) and English essay scoring. Researchers approach English essay scoring from rule-based approaches (Mitchell et al., 2002) and different machine learning approaches including regression (Page, 1968, 2003; Burstein, 2003; Cozma et al., 2018), classification (Larkey, 1998; Rudner and Liang, 2002; Rosé et al., 2003; Coniam, 2009) and ranking (Briscoe et al., 2010; Yannakoudakis et al., 2011; Chen and He, 2013) based models. More recent work focus on neural network without handle crafted feature engineering but innovation on model structure (Alikaniotis et al., 2016; Taghipour and Ng, 2016; Dong and Zhang, 2016; Dong et al., 2017; Tay et al., 2018). Other aspects have been addressed in ATS including spelling and grammatical errors (Leacock and Chodorow, 2003a; Han et al., 2006; Tetreault and Chodorow, 2008), semantics (Rehder et al., 1998; Lonsdale and Strong-Krause, 2003), coherence (Miltsakaki and Kukich, 2000, 2004; Higgins and Burstein, 2007; Burstein et al., 2010; Yannakoudakis and Briscoe, 2012; Burstein et al., 2013; Somasundaran et al., 2014), discourse (Burstein et al., 2001; Lin et al., 2011; Feng et al., 2014; Song et al., 2017), argumentation (Persing and Ng, 2015; Ghosh et al., 2016; Klebanov et al., 2016; Wachsmuth et al., 2017), prompt (Higgins et al., 2006; Persing and Ng, 2014; Cummins et al., 2016a; Rei, 2017), clarity (Persing and Ng, 2013), structure (Somasundaran et al., 2016; Persing et al., 2010), sentence parallelism (Song et al., 2016), question difficulty (Pado, 2017), non-English response detection (Yoon and Higgins, 2011), utilising data from different sources (Phandi et al., 2015; Cummins et al., 2016b) and evaluation (Williamson, 2009; Yannakoudakis and Cummins, 2015). In this section, we describe several influential approaches in this field. For more detailed reviews of ATS, see Shermis and Burstein (2003); Valenti et al. (2003); Pérez-Marín et al. (2009); Shermis and Burstein (2013);

Burrows et al. (2015).

Page is one of the pioneers of ATS research. The Project Essay Grade (PEG) (Page, 1968, 1994; Page and Petersen, 1995; Page, 2003) built by his team was the first attempt at building an ATS system. In the original paper (Page, 1968), Page claimed that the writing quality of an essay could be measured by different observable attributes. He proposed several features to approximate these attributes, which he named *proxes*. The proxes used in his system include:

- word count;
- the number of pronouns, POS tags, punctuation marks and paragraphs;
- word length describing the complexity of word usage;
- the presence of a title

Linear regression was used to learn the relationship between essays and scores.

Page's original work in the 1960s was not widely used because computer usage availability was minimal at the time, and Page needed to hire staff to transcribe texts from paper to tape and punched cards so that a computer could directly handle and read these texts (Page, 2003). This problem did not change until the 1990s when more families could afford and access computers. This not only encouraged more researchers to work on ATS, but also made it easier to build larger learner corpora and apply more advanced technology in ATS. For example, Shermis et al. (2001) added a web interface to PEG to make it easier and faster to collect learners' texts for further research, and Page (2003) added more features including different dictionaries and parsers. PEG was evaluated on 1,314 essays provided by the Educational Testing Service (ETS) from the Praxis test to assess applicants for teacher certifications. The average correlation between the system and human judge was 0.742, even higher than the average inter-annotator correlation 0.646 (Page and Petersen, 1995).

In terms of the real-world application of ATS systems, Kukich (2000) pointed out two problems that have discouraged people from using ATS systems. One problem is that these systems are relatively easy to fool. Specifically, since longer essays tend to get higher scores, learners could cheat simply by writing more words. Another more crucial problem is that people think these proxes do not capture other critical writing features including content, organisation and style. It may be artificial to emphasise the ability to score highly on an ATS system, when this might not correlate with the ability to plan, compose and reorganise high-quality texts in different conditions and circumstances.

To make an ATS system work better, different researchers have approached this problem from various angles. For example, Larkey (1998) brought classification approaches to ATS to add extra features to the regression model. He used the naive

Bayes classifier (Maron, 1961) to predict whether an incoming essay was good or bad, and the k-nearest-neighbour classifier (Altman, 1992) to gauge the similarity between the incoming essay and all the low-quality and well-written essays in the training set. For naive Bayes, he built several classifiers, one for each level band. Each classifier was used to predict the probability that the grade of an incoming essay was higher or lower than the given level band. Compared to the Project Essay Grade, Larkey also injected more lexical information into his ATS system via his classification models. A bag of stemmed words was used to represent the features of each essay in naive Bayes. For the k-nearest-neighbour classifier, they encoded each essay via its word occurrence weighted by term-frequency inverse-document-frequency (Spärck Jones, 1972). Finally, a linear regression grader with low variance features filtered was built to predict essay scores based on the outputs from the naive Bayes, the k-nearest-neighbour classifier and extra features measuring the length statistics of characters, words and sentences, and word varieties in each essay. The grader was trained and tested on five different datasets separately from social studies, physics, law and two general college exams, and Larkey experimentally showed that the classification models with lexical information could boost ATS system performance based on text-complexity features such as word length and the number of sentences in one essay.

Rudner and Liang (2002) treated ATS as a classification task as well. Their system, the Bayesian Essay Test Scoring sYstem (BETSY), was based on the multivariate Bernoulli model (McCallum and Nigam, 1998) and the multinomial naive Bayes model to classify a given essay with an appropriate grade. The features they used include the word unigram and bigram counts in each essay. They conducted their research on the responses to a biology exam for the Maryland High School Assessment. The best combination of model, features and feature selection method achieved 80% accuracy. One potential problem with their approach was that it ignored the order among different level bands. For example, given the three levels 1, 2 and 3 in a grading criterion, the classification method they used did not know that score 3 is better than 2 and 1, and 2 is better than 1. This is different from Larkey (1998)'s work, which preserved the order between bands. Furthermore, the likelihoods from Larkey's classifiers were also used as features in a regression model, which further preserved the difference between bands. Further experiments in Briscoe et al. (2010) also showed that the ranking and regression approaches outperformed treating ATS as an unordered classification task such as naive Bayes and maximum entropy classification.

Foltz et al. (1999) built their ATS grader to mark essays. Their product Intelligent Essay Assessor (IEA) mainly used the features from the following five categories:

- content (essay semantic similarity and vector length based on latent semantic analysis);

- lexical sophistication (word variety and confusable words);
- mechanics (punctuation, spelling and capitalisation);
- grammar (n-gram features, grammatical errors and grammar error types);
- style, organisation and development (sentence-sentence coherence, overall essay coherence and topic development)

Unlike the work we have discussed so far, they started to emphasise the importance of content using latent semantic analysis (LSA) (Deerwester et al., 1990; Laham and Landauer, 1998; Landauer et al., 1998). For LSA, they trained a semantic model from a large domain-specific background corpus. The texts from this corpus were similar to the reading materials read by the learners who participated in their exams. LSA builds a co-occurrence matrix A for the background corpus, in which the rows and columns represent the words and the paragraphs in the corpus, respectively. Element a_{ij} in the matrix A describes whether word i appears in paragraph j of the background corpus or not. The truncated singular-value decomposition is then applied to compress the matrix to a smaller one.

To use LSA to mark a text submitted by a learner, they treated each text as a bag of words (Harris, 1954), in which each text was a vector and each element of that vector was a word count. The bag of words vector for each text was then mapped to the latent space. They then calculated the cosine similarity between the model answers and learners' responses to represent the similarity of the conceptual content. For each learner's essay, they selected the top K graded essays with the highest cosine similarity as the learner's essay, and gave the learner's essay a score equal to the average of these top K essay grades. In subsequent work, they found that the length of the LSA vector strongly correlated with the essay scores (Rehder et al., 1998). In addition to semantic features, Foltz et al. (1998) used coherence features which were calculated from the average similarity of adjacent sentences in learners' texts according to LSA. Although they claimed other types of features, such as the strength of introduction, arguments and conclusion, were also used in their system, they did not disclose how these features were extracted and used in ATS.

The Educational Testing Service (ETS) developed their ATS system e-rater and deployed it to score high-stakes assessments including Graduate Record Examinations (GRE) and TOEFL (Burstein, 2003; Attali and Burstein, 2004). E-rater has several sub-systems that extract features from each text. The outputs from these sub-systems are then combined together. They grouped their features into the following categories:

- grammatical errors;
- discourse structure and organisation-related features;

- topic-relevant word usage;
- style-related word usage;
- sophistication, register and relevance of word usage.

The grammatical errors are detected by different error detection systems including erroneous bigrams (Leacock and Chodorow, 2003a), article errors (Han et al., 2006) and preposition errors (Tetreault and Chodorow, 2008). Discourse structure and organisation development features are extracted by building a discourse function classifier to detect the role of each sentence (Burstein et al., 2003). Topic-relevant word usage is measured by the word frequency cosine similarity between the incoming essay and the other score essays from the same topic (Burstein et al., 1998). Style-related word usage is measured in terms of whether a word is overly repeated (Burstein and Wolska, 2003). Sophistication features include the ratio between word type and token, vocabulary level, and average word length (Yigal and Burstein, 2006).

After the features are extracted from each text, a linear regression model is applied to predict the essay scores in an exam. The model is trained to learn the weights of some features above, while the weights of other features are pre-defined manually based on the requirements and goals of the exam. E-rater has also deployed an anomaly detection system to filter essays, which detects essays that might be either off topic, copies of the prompt, written in bad faith, a result of keyboard mashing, or otherwise exceptionally similar to other essays, hence likely to be plagiarised.

Briscoe et al. (2010) were the first to investigate ATS on the exams from Cambridge Assessment. In their work, they treated ATS as a ranking problem and proposed a discriminative model called Time Aggregated Perceptron (TAP), which is a variation of the batch perceptron method (Rosenblatt, 1958; Bös and Opper, 1998). During training, TAP iteratively updates the model parameters in the direction of all misclassified items. The step size for the parameters in each iteration is controlled by a decaying hyper-parameter. This decaying hyper-parameter causes the step size to gradually shrink, which terminates the parameters update. They used TAP in the context of pair-wise ranking, to determine which of the two texts written by two learners was better. The texts forming the pairs in the training set were sampled to reduce the number of data instances to train the ranking model. The distribution of the sampled training set was restricted to the same score distribution as the original training set. They showed that their TAP ranking model outperformed the classification approaches including naive Bayes and maximum entropy in this task. Moreover, they found a strong temporal effect in ATS. Specifically, they trained a model on data collected during different years and evaluated the model on the FCE exam papers collected in 2001. The model trained on the 400 essays, collected in 2001, achieved 0.69 in terms of the Pearson correlation, but only 0.60 if the model was trained on the data from the year 1997, even though the training set for that year was more than twice as large with 900 essays. Based on these results, they suggested that either the type of prompts or the marking rubrics of the exam or both might have evolved and changed over time, which contributed to this performance degradation.

Yannakoudakis et al. (2011) extended Briscoe et al. (2010)'s work by adding grammar complexity features. For each sentence in a learner's text, these complexity features measure the distance between any pair of words in the parse tree of this sentence. They believed that longer range syntactic dependencies written by a learner might demonstrate that the learner has a higher proficiency level in English writing. They released a subset of the FCE dataset used in Briscoe et al. (2010) as the FCE-PUB dataset. They also compared SVM ranking and support vector regression on the FCE-PUB dataset. The ranking model significantly outperformed support vector regression in terms of Pearson and Spearman correlations.

In their subsequent work, Yannakoudakis and Briscoe (2012) investigated how different coherence features affected the overall score of the texts in the FCE-PUB dataset. They showed that the coherence features based on incremental semantic analysis (ISA) (Baroni et al., 2007) measuring average adjacent sentences similarity improved their ATS system in terms of Pearson and Spearman correlations. Yannakoudakis (2013) combined their previous work (Yannakoudakis et al., 2011; Yannakoudakis and Briscoe, 2012) together and compared ATS in the FCE and IELTS exams. They found that the coherence features such as ISA working for one exam might not work for another exam, and vice versa. Consequently, coherence feature transferability across different exams in ATS still deserves further investigation.

Recent work in ATS investigated using deep neural network models on raw word features. Alikaniotis et al. (2016) showed that a bi-directional long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) neural network model with pre-trained word embeddings outperformed a model with feature engineering on the Automated Student Assessment Prize (ASAP) dataset (Section 2.1). Taghipour and Ng (2016) improved model performance of their neural network model by averaging the LSTM hidden layers across all timestamps. Dong et al. (2017) demonstrated that modelling the relationship between sentences in their neural network model could further improve model performance. Tay et al. (2018) added a coherence feature into their neural model. Although progress is visible in building deep neural network models for ATS, the performance of a model heavily depends on which datasets are evaluated. All the deep neural network work mentioned here was evaluated on the ASAP dataset. Farag et al. (2017) used the same model in Taghipour and Ng (2016) and tested their model on the FCE-PUB dataset. Even if Farag et al. used more training data and fine-tuned their embeddings on the error annotated Cambridge Learner Corpus (Section 2.1), there was still a performance gap between their neural model and the shallow model with feature engineering implemented by Yannakoudakis and Briscoe (2012). In addition, Cozma et al. (2018) showed that performance improvement could be achieved on the ASAP dataset with a shallow model based on a histogram intersection string kernel (Ionescu et al., 2014) and features derived from clustering embeddings (Butnaru and Ionescu, 2017). Therefore, we did not include any results from neural models in this thesis.

As we conducted our research on the exams from the Cambridge Assessment, and the most relevant ATS work on these exams we know is the work done by Briscoe et al. (2010); Yannakoudakis (2013) on the FCE exam. Therefore, we will build our baseline models and extend their work based on Briscoe et al. (2010); Yannakoudakis (2013) in this thesis.

2.5 Methods Differentiating Data with Different Meta-Properties

The grading criteria for texts answering one question might be different for another question, and thus training a single ATS system for texts, without discerning questions, might ignore the potential influence on scoring brought by their properties. For example, a high-quality letter should use correct salutations and closing phrases. In comparison, a text summarising a figure or table might rely on more numerical expressions to describe its details. Such exam-dependent properties discussed in Section 1.1 affect the way in which a piece of text is evaluated.

However, training separate ATS systems for texts answering different questions might not be an ideal approach, as texts with different properties may also share commonalities in their grading criteria. For example, we can see that in the exam products in Section 2.1, texts are penalised for misspellings and grammatical errors if they hinder reading comprehension. Therefore, if we were to isolate these texts to train different grading systems separately, each system would have fewer texts from which to learn the relationship between texts and scores, and this might lead to poorer performance relative to training a single and universal grader for the whole set of texts.

In summary, the differences and commonalities between the questions learners answer might be reflected in their corresponding grading criteria, and *multi-domain learning* is a technique to differentiate data from various domains, while capturing the similarities between domains. In this section, we summarise related work in multi-domain learning and research progress for multi-domain learning in ATS. We borrow the concepts from multi-domain learning (Joshi et al., 2012) to define the meta-properties categorise a text as the **attributes** of the text, in which each attribute can be used to categorise various texts into different **domains**.

2.5.1 Overview

We begin with a brief review of existing multi-domain learning techniques and related topics discussed in previous work. A more detailed summary can be found in the reviews written by Jiang (2008); Pan and Yang (2010); Weiss et al. (2016).

Multi-domain learning is highly relevant to domain adaptation, multi-task learning and transfer learning. Previous work has failed to reach a consensus on a universally accepted definition of these four paradigms, and the terminology used to describe them has often been mixed. Here we adopt the following definitions for the four paradigms:

- *domain adaptation* adapts knowledge from one or more source domains to improve model performance on the data from the target domain on the same task (Pan and Yang, 2010).

- *transfer learning* is similar to domain adaptation, but the process can relate to a different task (Pan and Yang, 2010). For example, it may involve adapting knowledge from tasks such as POS tagging, parsing or grammatical error correction to ATS. In transfer learning (Section 2.5.1), when we build a model to mark the texts from a single dataset, we only use texts from other datasets to optimise model performance on this single dataset. This single dataset is the *target dataset*; the other datasets excluding the target dataset are the *source datasets*.

- *multi-domain learning* emphasises improving the overall performance of data instances from all domains under the same task by modelling the similarities and differences between domains.

- *multi-task learning* aims at working on different tasks by sharing common knowledge, while separating and persevering the differences between tasks (Caruana, 1998). Multi-task learning emphasises multiple tasks, while multi-domain learning focuses on only one single task.

2.5.2 Related Methods

In this section, we briefly describe some common methods used in this area.

Daume III (2007) proposed frustratingly easy domain adaptation (FEDA), which is a feature-augmentation method used to model the similarities and differences between data from different domains. This method has a shared representation for all domains, and each domain duplicates the shared representation to capture domain-specific patterns useful in prediction as the domain-specific representation. Daume III showed that this method can improve model performance in named entity recognition, POS tagging and recapitalisation on data from different domains. More details will be described in Section 5.3.

Finkel and Manning (2009) proposed hierarchical Bayesian domain adaptation for named entity recognition and dependency parsing. In their framework, the authors grouped data instances from different domains into a hierarchy, in which each node corresponds to a conditional random field (CRF) model (Lafferty et al., 2001) for a domain. Each model shares the parameters inherited from their parent model by adding an L2 regularisation penalty multiplied by a hyper-parameter σ to force models to be similar to their parent model. All of the models in this hierarchy are trained together. At test time, the model at each leaf node is used to predict the labels of its corresponding data instances. Finkel and Manning demonstrated that their model is theoretically equivalent to FEDA, with extra hyper-parameters σ controlling the similarity between different domains.

Blitzer et al. (2006) described an domain adaptation method called *structural correspondence learning* (SCL). In their method, after extracting the feature vector $\mathbf{x}$ from each data instance x, they selected the top m most frequent features as the pivot features. For each pivot feature f, they trained a linear classifier with parameters $\mathbf{w}_f$ to predict whether feature f exists in each feature vector $\mathbf{x}$ to determine the correlation between features. Matrix W represents all the parameters $\langle \mathbf{w}_1, \ldots, \mathbf{w}_m \rangle$ learnt for each feature f, and matrix W is further factorised using singular value decomposition $W = UDV^T$ with $\theta = U^T_{[1:h,:]}$, with h representing the hyper-parameter used to determine the dimension of the projected space. This step creates a new representation for feature vector $\mathbf{x}$ in a projected space. Each feature vector $\mathbf{x}$ is then augmented with $\theta\mathbf{x}$ as $\mathbf{x} \oplus \theta\mathbf{x}$, and these augmented feature vectors are finally fed into a machine learning model used to learn the relationship between the augmented input space X^{aug} and the output space Y.

Jiang and Zhai (2007) proposed an instance weighting method for domain adaptation, applying different weights for instances from the source and target domains. They first trained a logistic regression classifier on the data from the target domain and filtered out the top instances from the source domain that were wrongly classified with the highest confidence, as this might constitute noise and negatively affect the classifier's performance on the target domain. They then used the trained classifier to obtain predictions for the unlabelled data instances from the target domain. The top instances with their predicted labels with the highest confidence were used to expand the training set in a style similar to self training and semi-supervised learning. Finally, they weighed all data instances from the target domain more highly than those from the source domain in order to make the model attend more closely to the target domain.

They experimentally demonstrated that their method could result in improvement for POS tagging, named entity recognition and spam filtering.

Dredze et al. (2010) introduced a multi-domain learning and domain adaptation framework based on the confidence-weighted linear classifier (Dredze et al., 2008). The confidence-weighted linear classifier is an online-learning classifier defined by a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ with a diagonal covariance matrix $\Sigma \in \mathbb{R}^{N \times N}$ and mean parameters $\boldsymbol{\mu} \in \mathbb{R}^N$. Σ also defines the confidence values over $\boldsymbol{\mu}$. More specifically, the ith diagonal item Σ_i in Σ determines how much uncertainty the classifier has in μ_i in $\boldsymbol{\mu}$. In their method, Z confidence-weighted linear classifiers, and each classifier is trained for the data instances from one domain z with Σ^z and $\boldsymbol{\mu}^z$. These classifiers are then combined to form a new classifier c^c with Σ^c and $\boldsymbol{\mu}^c$ by minimising the sum of the distances $\mathbb{D}$ between the combined classifier and all the classifiers for each domain:

$$(\boldsymbol{\mu}^c, \Sigma^c) = \arg\min_{\boldsymbol{\mu}^c, \Sigma^c} \sum_{z=1}^{Z} \mathbb{D}(\mathcal{N}(\boldsymbol{\mu}^c, \Sigma^c) || \mathcal{N}(\boldsymbol{\mu}^z, \Sigma^z); \boldsymbol{b}^z) \tag{2.25}$$

$\boldsymbol{b}^z = < b_1^z, \ldots, b_N^z > \in \mathbb{R}_+^N$ represents the hyper-parameters to control the contribution of the classifier trained for domain z. $\mathbb{D}$ represents the distance between two distributions. As $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ is a multivariate Gaussian distribution with a diagonal covariance matrix, the authors further defined $\mathbb{D}(\mathcal{N}(\boldsymbol{\mu}^c, \Sigma^c) || \mathcal{N}(\boldsymbol{\mu}^z, \Sigma^z); \boldsymbol{b}^z)$ as the weighted sum of multiple univariate Gaussian distributions:

$$\mathbb{D}(\mathcal{N}(\boldsymbol{\mu}^c, \Sigma^c) || \mathcal{N}(\boldsymbol{\mu}^z, \Sigma^z); \boldsymbol{b}^z) = \sum_{i=1}^{N} b_i^z \mathbb{D}(\mathcal{N}(\mu_i^c, \Sigma_i^c) || \mathcal{N}(\mu_i^z, \Sigma_i^z)) \tag{2.26}$$

The distance $\mathbb{D}(\mathcal{N}(\mu_i^c, \Sigma_i^c) || \mathcal{N}(\mu_i^z, \Sigma_i^z))$ between two Gaussian distributions could be the L2 distance defined over the differences of means and variances of the distributions:

$$\mathbb{D}_{L2}(\mathcal{N}(\mu_i^c, \Sigma_i^c) || \mathcal{N}(\mu_i^z, \Sigma_i^z)) = (\mu_i^c - \mu_i^z)^2 + (\Sigma_i^c - \Sigma_i^z)^2 \tag{2.27}$$

and the optimal values for μ_i^c and Σ_i^c by setting the derivative of $\mathbb{D}_{L2}$ over μ_i^c and Σ_i^c to 0 will be:

$$\mu_i^c = \sum_{z=1}^{Z} \tilde{b}_i^z \mu_i^z \tag{2.28}$$

$$\Sigma_i^c = \sum_{z=1}^{Z} \tilde{b}_i^z \Sigma_i^z \tag{2.29}$$

and $\tilde{b}_i^z = \frac{b_i^z}{\sum_{z=1}^{Z} b_i^z}$. $\mathbb{D}$ could also be the Kullback-Leibler divergence $\mathbb{D}_{KL}$ between two probability distributions:

$$\mathbb{D}_{KL}(\mathcal{N}(\mu_i^c, \Sigma_i^c)||\mathcal{N}(\mu_i^z, \Sigma_i^z)) = \frac{1}{2}(log\frac{\Sigma_i^z}{\Sigma_i^c} + \frac{\Sigma_i^z + (\mu_i^c - \mu_i^z)^2}{\Sigma_i^c} - 1) \tag{2.30}$$

and the optimal values yield:

$$\mu_i^c = \left(\sum_{z=1}^{Z} \frac{\tilde{b}_i^z}{\Sigma_i^z}\right)^{-1} \sum_{z=1}^{Z} \frac{\tilde{b}_i^z}{\Sigma_i^z}\mu_i^z \tag{2.31}$$

$$\Sigma_i^c = \left(\sum_{z=1}^{Z} \frac{\tilde{b}_i^z}{\Sigma_i^z}\right)^{-1} \tag{2.32}$$

The combined model is finally used to predict the label of a data instance $\mathbf{x}_i$ from a domain as $sign(\boldsymbol{\mu}^c \cdot \mathbf{x}_i)$. The authors showed that their multi-domain learning models performed better than the baselines without discerning domain differences and FEDA on spam filtering and sentiment analysis. However, their method is only applicable to classification, which does not fit to this thesis.

2.5.3 Distinguishing Attributes in ATS

Some studies have aimed at distinguishing attributes in ATS.

Heilman and Madnani (2013) studied multi-domain learning for automated short answer scoring on two datasets. Their first dataset was the Beetle dataset (Dzikovska et al., 2012), which focuses on basic electricity and electronics, while the second dataset was the Science Entailment Corpus (Nielsen et al., 2008), which covers a wide range of scientific topics. The authors scored each answer using FEDA that distinguishes which dataset and question each answer corresponded to. They only examined short answer scoring on scientific topics, while in this thesis, we will look at ATS for English as a foreign language (EFL) learners. Furthermore, they did not weigh the contributions from various domains differently, and their datasets included only one type of question, while Finkel and Manning (2009) showed that weighting domains differently can improve model performance.

Phandi et al. (2015) studied domain adaptation using the Automated Student Assessment Prize (ASAP) dataset described in Section 2.1. In their work, the authors split the essays from eight prompts in the ASAP dataset into four folds. Each fold contained the essays from two prompts. They then examined how the essays from one prompt could improve model performance on the essays for the other prompt in each fold. They modified Bayesian linear ridge regression (Bishop, 2006, p.152) by adding a prior so their model could dynamically learn how much contribution to derive from the source domain to assist the grading system on the target domain. Their work only studied the adaptation of essays from one prompt to another, and they only studied prompts in their work.

Cummins et al. (2016b) approached the ATS problem using a multi-task learning framework. Similar to Phandi et al. (2015), they also worked on the ASAP dataset. The authors used FEDA to encode each prompt, then built a timed aggregate perceptron (Briscoe et al., 2010) to rank any pair of essays on quality. Finally, they trained a linear regression model to map the ranking scores for each prompt to the original scores. Similar to Phandi et al., they only studied prompt. Besides that, they only weighed each prompt equally in FEDA, which is similar to what Heilman and Madnani (2013) have done.

In summary, the work utilising meta-properties in ATS focused on texts written by English native speakers with FEDA and its variant, and there might be some useful meta-properties not identified in previous work (e.g. genre). To supplement the missing parts of previous research, we are studying the texts written by EFL, and we aim to identify more intra-exam meta-properties that might be helpful in ATS in Chapter 5 and investigate inter-exam meta-property for exams targeting different English proficiency levels in Chapter 6. We also want to study how multi-domain learning and transfer learning can help us improve model performance for the datasets provided by Cambridge Assessment. For the multi-domain learning and transfer-learning techniques, we start with FEDA, because it is a well-tested and easily-implemented method, proven to be useful in ATS. More specifically, in Chapter 5, we will use FEDA to differentiate multiple intra-exam meta-properties and use the FEDA variant to weigh meta-properties differently. In Chapter 6, we will use transfer learning and instance weighting to leverage more datasets for which the grading criteria are hugely different. In future work, we might expand our work to more extreme cases such as when we have limited data for the target domain, and in this scenario, we can test methods such as structure correspondence learning to explore useful features in ATS.

2.6 Evaluation

In order to measure how well an ATS system performs, we need a suitable evaluation procedure. In ATS evaluation, researchers collect a group of texts marked by trained human judges. We then compare the scores predicted by an ATS system against these gold human scores in order to calculate various quantitative metrics, and these metrics quantitatively describe model performance. This evaluation procedure gives us consistent and instant feedback on how well an ATS system performs.

Previous researchers (Chen and He, 2013; Alikaniotis et al., 2016; Taghipour and

Ng, 2016; Dong and Zhang, 2016; Dong et al., 2017; Tay et al., 2018; Cozma et al., 2018), investigating the ASAP dataset, reported quadratic weighted Cohen's kappa (Cohen, 1960, 1968). Briscoe et al. (2010); Yannakoudakis et al. (2011); Yannakoudakis and Briscoe (2012); Alikaniotis et al. (2016) reported correlation metrics such as Pearson (Pearson, 1895) and Spearman (Spearman, 1904) correlations. Alikaniotis et al. (2016); Persing and Ng (2013, 2014, 2015) also included error metrics such as mean-absolute error and root-mean-squared error (RMSE) in their results.

Yannakoudakis and Cummins (2015) summarised and discussed the problems behind various metrics used in ATS. The authors went against Cohen's kappa in ATS, and one of the reasons they suggested was the bias problem in Cohen's kappa (Byrt et al., 1993; Eugenio and Glass, 2004; Sim and Wright, 2005; Powers, 2012). In the bias problem, Cohen's kappa rewards an ATS system whose predicted scores marginal distribution is different from gold scores, even if the percent agreement between the predictions and gold scores gets lower (Graham and Jackson, 1993), and vice versa (Feinstein and Cicchetti, 1990; Cicchetti and Feinstein, 1990). Brennan and Prediger (1981) suggested that it is appropriate to use Cohen's kappa only if both marginals to be compared in kappa are fixed, which does not apply in most machine learning evaluation scenarios, because the marginal of system predictions can freely vary. Furthermore, Uebersax (1987); Gwet (2002); Pontius Jr and Millones (2011) criticised the agreement correction in Cohen's kappa, which describes a random-guessing baseline following the gold score marginal distribution. It might be misleading as most systems do not make predictions in this way. More discussion related to Cohen's kappa can be found in Craggs and Wood (2005); Artstein and Poesio (2008).

Yannakoudakis and Cummins (2015) also criticised the usage of correlation metrics, because these metrics do not take any systematic biases into account. In other words, the correlation between gold scores and system predictions could still be +1 if system predictions are consistently higher or lower than gold scores. Another problem with Spearman's correlation is that it does not handle ties well, despite attempts to remedy this problem (Amerise and Tarsitano, 2015). According to the grading scale of the exams in Cambridge English, CAE and FCE are marked based on four different aspects (to be described in Section 3.2), each of which has five different levels. If we aggregate the scores from these fours aspects as the total scores, then there are 21 different scores if we also include 0. As more responses are collected for an exam, the number of ties increases rapidly, so this problem cannot be ignored.

In this thesis, we will optimise our model based on RMSE, and RMSE is the major metric we are going to investigate. Compared to Cohen's kappa, RMSE does not suffer from the bias problem, and neither does it have the problems discussed in the correlation metrics. To benchmark and facilitate comparison with previous works, we also report the Pearson and Spearman correlations for reference. We do this because our work is closely related to the work of Briscoe et al. (2010); Yannakoudakis et al. (2011); Yannakoudakis and Briscoe (2012) (Section 2.4), because they mainly reported results in terms of Pearson and Spearman correlations, we will do the same for reasons of comparison.

Chapter 3

Datasets and Baseline

To study ATS, we first need to determine the datasets and ATS model we wish to investigate. In this chapter, we describe the datasets and baseline model evaluated in this thesis. The baseline model is an extended version proposed by Yannakoudakis et al. (2011), which we use as a benchmark for our modified models in the following chapters.

3.1 Exams

In this section, we first describe the exams from which we collected our data. All the exams were designed by Cambridge Assessment. More Specifically, our work focuses on the main suite and the International English Language Testing System (IELTS) exam described in Section 2.1.2.1. Each exam in the main suite has a target CEFR proficiency level, while IELTS is designed to evaluate learners' skills at all proficiency levels. The main suite consists of:

- the Key English Test (KET);
- the Preliminary English Test (PET);
- the First Certificate in English (FCE);
- the Certificate in Advanced English (CAE);
- the Certificate of Proficiency in English (CPE).

For the main suite, the exams we have used in this thesis exclude CPE as we lack sufficient data for this exam. The target CEFR proficiency level for each exam is presented in Figure 3.1.

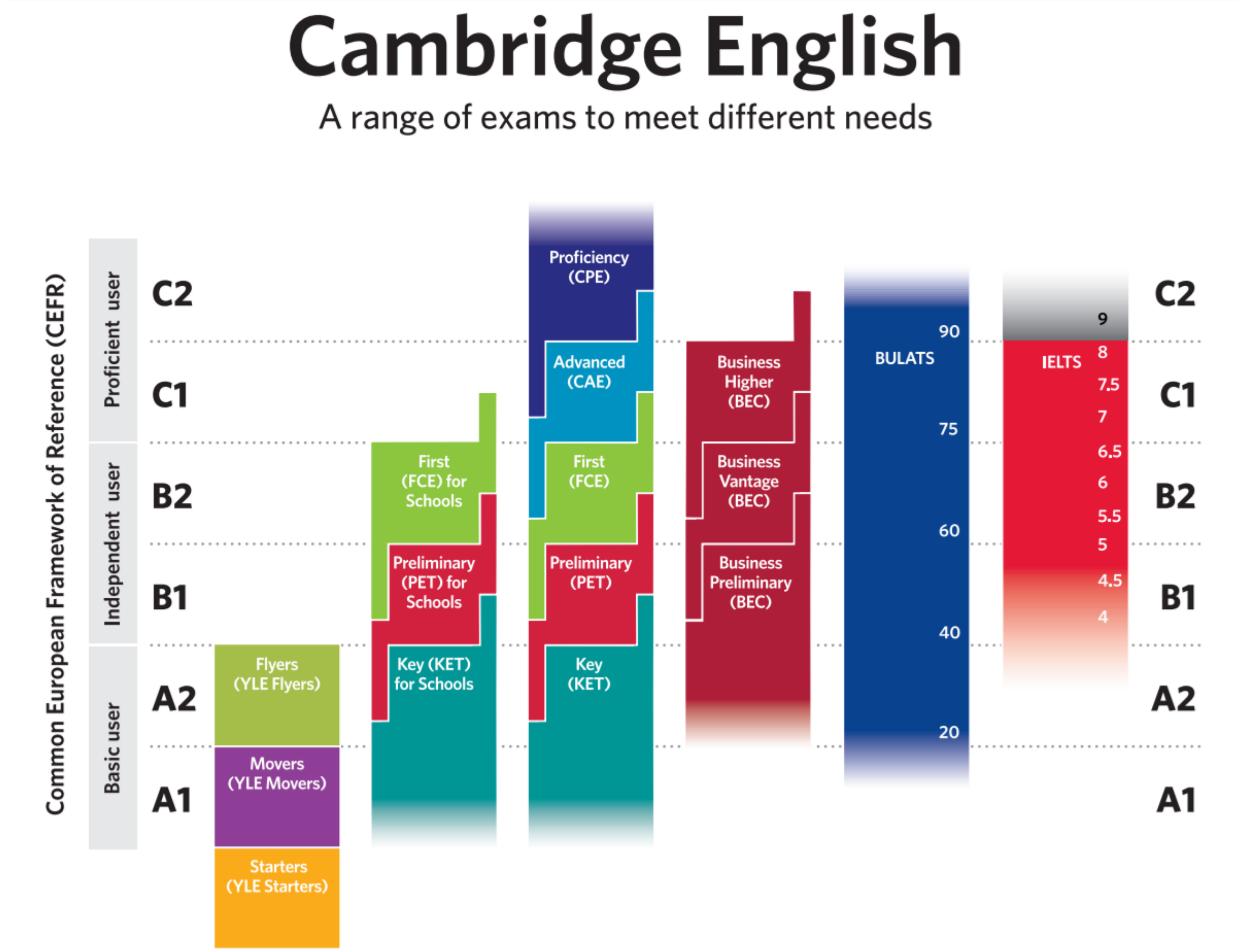


Figure 3.1: Interpretation between each exam to its CEFR level

3.1.1 KET

KET is used to assess learners' abilities to use English in simple situations.[1] Its target CEFR level is A2. The KET writing section only contains one writing task: composing a short message based on given instructions. The number of words required is 25-35.

3.1.2 PET

PET is used to assess learners' mastery of English basics.[2] Its target CEFR level is B1. The PET writing section contains continuous writing. Learners must choose between writing an informal letter or a story based on two given questions. In either response, they are required to write approximately 100 words.

[1] www.cambridgeenglish.org/exams/key/

[2] http://www.cambridgeenglish.org/exams/preliminary/

3.1.3 FCE

FCE is used to assess whether learners have the language skills to live and work independently in an English-speaking country or study on courses taught in English.[3] Its target CEFR level is B2. The exam contains two writing tasks. In the latest version of FCE by 2017, both writing tasks require learners to compose text passages of 140-190 words. The first writing task is a compulsory task that asks learners to write a letter based on a given prompt. Learners need to cover all the key information required in the prompt. In the second writing task, learners choose to write a report, a review or an article. Prior to 2015, learners could also choose to write a story in the second writing task.

There is one FCE public dataset released by Yannakoudakis et al. (2011), which was collected between 2000 and 2001. At that time, the FCE exam had a different word count requirement (120-180). In this thesis, we use **FCE-PUB** to differentiate between the dataset released by Yannakoudakis et al. and the other FCE dataset we will use in this thesis. We will describe the latter in Section 3.3.

3.1.4 CAE

CAE is used to determine whether learners have reached a high level of achievement in learning, using and mastering English.[4] The exam is comprised of two writing tasks. The first task is compulsory, and it requires learners to write an essay describing their opinions to a given question. The second task allows learners to respond to one of three questions, and the three questions involve writing a letter/email, proposal, report or review depending on the given context, topic purpose and target readers.[5] Learners are required to finish both writing sections in 90 minutes, and the required word count for each task is 220-260 words.

3.1.5 IELTS

IELTS is used to test the language skills of learners wishing to study, work or live in an English-speaking country. This exam can be further classified into two categories: IELTS academic and IELTS general training. Both exams are comprised of two required writing tasks.[6] The first task in the IELTS academic exam presents learners with a graph, a table, a chart or a diagram to describe, summarise and explain the key information conveyed. In contrast, the first task in the IELTS general exam presents a situation and

[3] http://www.cambridgeenglish.org/exams/first/

[4] http://www.cambridgeenglish.org/exams/advanced/

[5] http://www.cambridgeenglish.org/exams/advanced/exam-format/

[6] https://takeielts.britishcouncil.org/prepare-test/understand-test-format/writing-test

asks them to write a letter based on the situation, using an appropriate writing style. In both exams, learners are asked to write at least 150 words in their responses to the first task. The second writing task in both exams requires learners to write an essay that discusses an argument to express their own opinions , using at least 250 words. The writing time allocated in both exams is 60 minutes.

3.2 Grading Criteria

For the exams in the main suite, the KET writing task is evaluated by a single holistic score ranging from 0 to 5.[7] For PET, FCE and CAE, each answer is evaluated along four dimensions: content, communicative achievement, organisation and language. Each dimension is scored from Band 0 to Band 5.[8,9,10] In contrast, the FCE-PUB dataset, which refers to an older version of the FCE exam, uses single holistic score ranging from 0,0 to 5,3 (The *original* scores in Table 3.1). In the IELTS exam, both the first and second tasks are evaluated along four dimensions: both tasks are evaluated in the same three dimensions including coherence and cohesion, lexical resource, and grammatical range and accuracy. For the fourth dimension, the first task is evaluated according to task response, while the second task is for task achievement[11]. In addition, in both writing tasks, each dimension is scored from Band 0 to Band 9, and the scores from all four dimensions are averaged and rounded to 0.5 to represent the overall score of a text.

3.3 Dataset Preparation for Each Exam

As described in Section 2.2, in order to use a machine learning model to predict text scores, we require a training set for the model to learn the patterns between the input text space X and the output score space Y, a development set to tune the model hyperparameters, and a test set to evaluate model performance. Therefore, we split the KET, PET, FCE, CAE, IELTS and FCE-PUB datasets into the training, development and test sets. We use KET-PRI, PET-PRI, FCE-PRI and CAE-PRI to represent the datasets for the KET, PET, FCE and CAE exams, respectively.

For the FCE-PUB dataset, we follow the way used by Yannakoudakis et al. (2011), using all the 97 scripts from 2001 as the test set. Here, we borrow the term **script** used by

[7] http://assets.cambridge.org/97805217/54804/excerpt/9780521754804_excerpt.pdf

[8] http://www.cambridgeenglish.org/images/cambridge-english-assessing-writing-performance-at-level-c1.pdf

[9] http://www.cambridgeenglish.org/images/cambridge-english-assessing-writing-performance-at-level-b2.pdf

[10] http://www.cambridgeenglish.org/images/231794-cambridge-english-assessing-writing-performance-at-level-b1.pdf

[11] https://www.ielts.org/en-us/ielts-for-organisations/ielts-scoring-in-detail

Yannakoudakis et al. to refer to the set of texts written by a learner in one exam session. In each dataset, each script consists of one or two texts, and each text corresponds to a writing task. We then randomly split the scripts from 2000 into the training and development sets. For the other five datasets, we randomly sample 300 scripts for the test set and 300 scripts for the development set, and the rest for the training set.

3.3.1 Scores

Next, we choose the score for each text that we use in the output space Y for our ATS model.

For the KET-PRI dataset, we use the single holistic score for each text as the score for the ATS model to predict. All texts from the PET-PRI, FCE-PRI and CAE-PRI datasets are evaluated according to four aspects, and we add the scores of these four aspects together to obtain a total score in the range 0-20. We use this overall score as the output score for our study. As the original score of each text in the FCE-PUB was not reported on a numerical scale, Cambridge Assessment helps us convert the scores to integers between 0-20. The mapping is shown in Table 3.1. Finally, for the IELTS dataset, we directly use the existing total score as the individual score for each text for our study.

Original → New	Original → New
0,0 → 0	3,2 → 13
1,1 → 1	3,3 → 14
1,2 → 4	4,1 → 15
1,3 → 7	4,2 → 16
2,1 → 9	4,3 → 17
2,2 → 10	5,1 → 18
2,3 → 11	5,2 → 19
3,1 → 12	5,3 → 20

Table 3.1: Score mapping for the FCE-PUB dataset.

Some exam scripts referred to multiple tasks. Each task was given a single score, and the script was also given an overall score. To distinguish the scores here, we define the single score for each text marked by human examiners as the **individual-level score**, and the aggregate of these scores as the **overall-level score** for the script, reflecting the learner's writing overall proficiency level. Furthermore, in this thesis, we define the ATS model predicting individual text scores as the **individual-level model**, and the model predicting the overall-level score of of a single learner in one exam session as the **overall-level model**. In this dissertation, we mainly investigate the individual-level scores and potential improvements to the individual-level model, as the individual-level provide learners with more feedback on each text, relative to the overall-level scores.

We keep only those scripts with valid scores across all answer(s), because there are typos in some answers scores, and some answers do not have any score marked by any human examiner. Table 3.2 presents the statistics of each dataset.[12]

Dataset	CEFR	Score Range	MEAN	STD	#P	#S	#train	#dev	#test
KET-PRI	A2	0-5	3.86	1.32	6	2096	1496	300	300
PET-PRI	B1	0-20	15.26	2.40	16	2069	1469	300	300
FCE-PRI	B2	0-20	14.51	2.18	37	2047	1447	300	300
CAE-PRI	C1	0-20	13.20	2.69	50	2088	1488	300	300
IELTS	A1-C2	0-9	5.78	0.96	58	1604	1004	300	300
FCE-PUB	B2	0-20	13.92	2.92	31	1212	822	293	97

Table 3.2: Statistics of each dataset. MEAN and STD represent the mean and standard deviation of the scores in each dataset, respectively. #P and #S refer to the number of prompts and scripts, respectively.

3.4 Baseline Model

In this section, we present the baseline model, which is trained and evaluated using the datasets reported in Section 3.3.

3.4.1 Notation

We introduce some notation to facilitate our discussion. Each dataset consists of M task(s) for each learner to answer, and there are J learners for each dataset. We assume that each learner only takes any exam once. $t_{m,j}$ denotes the m^{th} text written by the j^{th} learner l_j, answering the m^{th} task $task_m$ in an exam session. Text $t_{m,j}$ can be represented as a sequence of words written by learner l_j. The individual score for text $t_{m,j}$ marked by a human examiner is $s_{m,j}$.

$TL_j = \{t_{m,j} | m = 1, ..., M\}$ denotes the set of all the texts written by learner l_j in an exam. In other words, TL_j is equivalent to the script answered by learner l_j in one exam session.

$TT_m = \{t_{m,j} | j = 1, ..., J\}$ denotes the sequence of all responses produced for the m^{th} task t_m answered by all learners in the same exam.

3.4.2 Features

In the baseline model, a feature vector $\mathbf{f}_{m,j}$ is extracted from text $t_{m,j}$. $\mathbf{f}_{m,j}$ is then used to train an individual-level model to learn the relationship between the feature vector

[12] The FCE-PUB dataset is publicly available at https://ilexir.co.uk/datasets/index.html (Yannakoudakis et al., 2011), while the other datasets have a restricted access within our research group.

space F and the text score space S. Ultimately, the model predicts the score of text $t_{m,j}$ as $\hat{s}_{m,j}$. In some cases, the predicted score $\hat{s}_{m,j}$ might be invalid on the given grading scale. For example, an ATS model might predict a score of 4.3, but it is invalid for a grading scale that requires an integer. Hence, we round each $\hat{s}_{m,j}$ to the nearest valid score on the given grading scale as $\hat{rs}_{m,j}$ (4 in the given example).

The baseline model we use is similar to that of Yannakoudakis et al. (2011) (Section 2.4). Specifically, both models are identical in terms of the following features:

1. the number of words in each text;
2. the count of phrase structure (PS) rules;
3. the trigram missing rate counting how many trigrams in each text are not covered in a background corpus, which are treated as erroneous trigrams;
4. the count of word unigrams and bigrams;
5. the count of POS unigrams, bigrams and trigrams based on the Constituent Likelihood Automatic Word-tagging System (CLAWS, Garside (1987)) C2 tagset.

However, there are also some differences between our model and that of Yannakoudakis et al.. Our model only uses the top parse result for grammatical relation distance measures, while they model used the top ten results. In addition, the trigram missing rate is only estimated on the UKWaC corpus (Ferraresi et al., 2008), whereas they also used the Cambridge Learner Corpus (Section 2.1) for this purpose. For the missing rate, we also count how many unigrams and bigrams are not covered in the background corpus.

In addition, we also include the following features:

- the number of misspelt words;
- the count of different grammatical relation types;
- the minimum, maximum and average sentence and word lengths.

We use the Moby lexicon[13] as a dictionary to check word spelling. Some learners wrote texts such as phone numbers, and these words, while correctly spelled, are not included in the dictionary. Furthermore, as the RASP tokeniser is originally optimised for parsing, some words correctly spelled by learners might not be optimally tokenised, so these tokens may not be found in the dictionary, either. Thus, we define a word as correctly spelled when:

- it contains any number, or

[13] /usr/share/dict/linux.words in Scientific Linux 6.8

- it appears in the dictionary when all punctuation marks are removed in this word (e.g. u.s.a → usa, append. → append), or
- the word is split into sub-tokens at each punctuation mark, and all subsequent tokens exist in the dictionary (e.g. day.Good → day Good)

Otherwise, we treat this word as a misspelt word.

The POS tags, grammatical relations and phrase structure rules described above are derived from the robust accurate statistical parsing (RASP) toolkit (Briscoe et al., 2006). We remove all features with a frequency in the training set of below four. The count of word unigrams and bigrams and POS unigrams, bigrams and trigrams are weighted using term frequency-inverse document frequency (TF-IDF) (Spärck Jones, 1972).. We then keep the top 25,000 features with the highest absolute Pearson correlation with the individual-level scores. Finally, each feature vector is normalised so that $||\mathbf{f}_{m,j}||_2 = 1$.

3.4.3 Models

In this thesis, we use the support vector regression (SVR) and ranking SVM implementation from LIBLINEAR (Fan et al., 2008) with L2 loss. The reason we use a linear kernel but not others is that Yannakoudakis et al. (2011) experimentally demonstrated that the ranking SVM (Section 2.3.3) with a linear kernel can produce competitive performance. In addition, it provides an intuitive explanation for the importance of each feature, which makes it easier for us to study the behaviour of the model we train. Furthermore, LIBLINEAR is highly optimised for the linear kernel, which is much faster than other SVM implementations, such as LIBSVM (Chang and Lin, 2011) and SVM$^{\text{light}}$ (Joachims, 2008).

3.4.4 Benchmark

Yannakoudakis et al. (2011) only built an overall-level model and evaluated it in terms of the Pearson (ρ_{prs}) and Spearman (ρ_{spr}) correlations on the FCE-PUB dataset. As we used more features and conducted feature selection based on Pearson's correlation, in this section, we benchmark our models with their results on the overall level.

We first concatenate all the texts in script TL_j to create the **concatenated text** ct_j so that

$$ct_j := t_{1,j} \oplus t_{2,j} \oplus ... \oplus t_{M,j} \tag{3.1}$$

We then extract the script feature vector cf_j from the concatenated text ct_j based on the features defined in Section 3.4.2. We define the combined script score cs_j of a script on the overall level as the sum of all the individual text scores within the script $cs_j := \sum_{m=1}^{M} s_{m,j}$

The FCE-PUB dataset has another overall script score ss_j for script TL_j, which is slightly different from the combined script score cs_j. In order to compare with Yannakoudakis et al.'s work, we build two overall-level models based on SVR and the ranking SVM respectively, and each overall-level model uses the script feature vector cf_j and its script score ss_j (rather than cs_j) with a linear kernel. For the ranking SVM model, in order to generate a valid predicted score on the given grading scale, we train another linear regression model $R^1 \rightarrow R^1$ to linearly fit the ranking scores predicted by the ranking SVM back to original overall-level scores. Finally, for each overall-level model, we round the scores that are predicted from its corresponding regressor to the nearest valid scores on the given grading scale.

We tune the cost hyper-parameter on the development set and report the results with the lowest RMSE on the development set. Results are presented in Table 3.3, which also reports RMSE, and the Pearson (ρ_{prs}) and Spearman correlation (ρ_{spr}) values. The upper part of the table shows previous results. UKWaC and CLC are the results reported in Yannakoudakis et al. (2011) on the ranking SVM models which use the UKWaC (Ferraresi et al., 2008) and the Cambridge Learner Corpus (CLC) (Nicholls, 2003) as the background corpus for n-gram missing rate estimation respectively. DISCOURSE is the CLC version with extra discourse features. In the DISCOURSE version, Yannakoudakis and Briscoe (2012) used the feature based on incremental semantic analysis (Baroni et al., 2007) measuring average adjacent sentence similarity to model the coherence of a text.

Table 3.3 does not include any recent neural model on the FCE-PUB dataset, because the neural model developed by Farag et al. (2017) showed that there is still a performance gap between the neural model and the models built on hand-crafted features for this dataset.

Model	RMSE	ρ_{prs}	ρ_{spr}
UKWaC	-	0.735	0.758
CLC	-	0.741	0.773
DISCOURSE	-	0.749	**0.790**
SVR	**3.988**	**0.761**	0.787
RANKING SVM + LINEAR FITTING	4.123	0.735	0.766

Table 3.3: Comparison of previous work and our baseline models for the FCE-PUB test set on the overall level.

As Table 3.3 shows, our models can achieve relatively good performance. The Pearson correlation ρ_{prs} of SVR is higher than that of DISCOURSE (0.761 versus 0.749), while the Spearman correlation ρ_{spr} is close (0.787 versus 0.790). We also found that by selecting appropriate features and hyper-parameters, the regression model can outperform the ranking model for ATS. This is different from Yannakoudakis et al.

(2011)'s finding that the ranking model is significantly better than the regression model for this task. The main reason we suggest might be the hyper-parameters selection. The default cost hyper-parameter used in Yannakoudakis et al. (2011) is 1.0 in SVM$^{\text{light}}$ if all the input vectors are L2 normalised, and the optimal cost hyper-parameter we found in this section is 2000.0, which might explain the difference between our findings and theirs. As we are optimising RMSE, and SVR achieves better performance on RMSE (3.988), we therefore use SVR in all the remaining experiments that are presented in this thesis.

3.5 Results

In this section, we present the results for each dataset evaluated on SVR for each dataset in Table 3.4. In this table, columns dev_rmse to dev_spr record the performance of SVR in terms of RMSE, the Pearson correlation (ρ_{prs}) and the Spearman correlation (ρ_{spr}) on each development set, and columns test_rmse to test_spr are the performance of SVR on the corresponding test set.

Dataset	dev_rmse	dev_prs	dev_spr	test_rmse	test_prs	test_spr
KET-PRI	0.970	0.661	0.585	0.968	0.706	0.648
PET-PRI	2.247	0.594	0.541	2.179	0.567	0.467
FCE-PRI	1.994	0.379	0.363	1.991	0.359	0.339
CAE-PRI	2.421	0.377	0.341	2.405	0.411	0.410
IELTS	0.701	0.697	0.680	0.693	0.684	0.659
FCE-PUB	2.402	0.567	0.571	2.569	0.662	0.652

Table 3.4: Results of the baseline model for each dataset on the individual level.

It is difficult to compare the same grader performance across datasets, because the datasets might differ in terms of their inherent properties or score distributions. However, if we naively conduct a comparison across of the absolute values, we can see that the model achieves better performance on some datasets (e.g. KET-PRI and IELTS), relative to others (e.g. FCE-PRI and CAE-PRI). One possible reason is that the features that work well on one dataset might not perform well for other datasets, which has been confirmed in previous work (Yannakoudakis, 2013) (Section 2.4). However, identification of the best feature set for each dataset is not the main goal of this thesis, so we will leave this as a future research direction.

3.6 Summary

In this chapter, we presented the datasets that we will use to conduct our research. We then described the implementation of a baseline automated text grader. The baseline model we built gets competitive performance compared to previous work. We finally evaluated this model using multiple datasets. The performance of the baseline system will be used to assess our models later on in the following chapters.

Chapter 4

Multi-Source Scoring by Bringing Authorship Knowledge

4.1 Introduction

To evaluate a learner's writing skill more thoroughly, many English exams such as IELTS and TOEFL ask them to answer multiple writing tasks. These tasks are drawn from different topics and genres, and each learner is required to write a text for each task. In practice, human judges score each text with an individual-level score, and these scores are aggregated to obtain an overall-level score, which reflects their writing skills.

When an individual-level ATS model scores texts, previous work has made an implicit assumption that all responses to all tasks are composed independently. This is not true for exams requiring responses to multiple tasks. The writing skill exemplified by a learner during the same exam session will not normally vary greatly, so the texts written by one learner may share some commonalities, such as preferential word usage and common mistakes, and should also approximately equally reflect their writing skills. We suggest that when an individual-level model predicts the score of a learner's text, it is helpful to use their other texts as a reference and pass it as an extra piece of information to the model. We refer to this information as **authorship knowledge**, and we define using multiple texts from the same learner as an extra reference to assess each individual text as **multi-source scoring**.

No previous work has investigated the utility of authorship knowledge in ATS except for Yannakoudakis et al. (2011). One possible reason is that most datasets only have one text written by each learner except for the First Certificate in English public (FCE-PUB) dataset in Section 3.1.3 released by Yannakoudakis et al., which contains two texts per learner. Yannakoudakis et al. did not explicitly consider the usage of authorship knowledge but built an overall-level model to directly predict the overall-level score of all the texts written by each learner. In this chapter, we will explicitly investigate the influence of authorship knowledge.

The potential benefit of passing this authorship knowledge to an ATS model might come from a reduction of data sparsity and an improvement in the robustness and reliability of feature extraction. Normally the text length to each task is limited, and so there may be insufficient features exemplified in a single response to differentiate the language quality of the response. Hence, it can be challenging for an ATS model to learn the mapping between texts and scores accurately.

In this chapter, we test the hypothesis that authorship knowledge can improve individual-level model performance. We pass this authorship knowledge to an individual-level model in two independent ways: feature fusion and score fusion. When the model predicts the score for one text, both methods pass all the texts written by the same learner to the model as an extra reference. We show that adding this knowledge could be helpful in an individual-level ATS model in some cases. To our knowledge, this is the first study that investigates how authorship knowledge affects the ATS system performance.[1]

4.2 Datasets

As the aim of this chapter is to investigate the effect of authorship knowledge on ATS, we require a dataset that includes more than one text written by each learner, and each text is scored with an individual score. We use the four datasets for our experiments described in Chapter 3 including FCE-PUB, IELTS, FCE-PRI and CAE-PRI. All datasets contain two writing tasks for learners to answer. The answers to both tasks were scored on the same grading scale. Each script was written on the same day so we can safely assume no dramatic variation in writing skill for each learner.

One problem that exists in these four datasets is that all the texts written by the same learner were marked by the same human examiner. More specifically, each human examiner marked the first and second text written by a learner in sequence. Therefore, the score of the second text might be affected by the first marked text, even though the grading criteria for each exam requested that each text should be scored independently. To wipe out this bias possibility, Cambridge Assessment provided extra two datasets for our research in this chapter. The extra two datasets FCE-S and CAE-S were collected from the FCE and CAE exams respectively after the year 2014. The difference between these two datasets and the previous datasets in Chapter 3 is that during the marking procedure, all the texts written by all learners in FCE-S and CAE-S were shuffled together.

[1] A version of this chapter has been published in the 13th Workshop on the Innovative Use of NLP for Building Educational Applications, North American Chapter of the Association for Computational Linguistics: Human-Language Technologies (Zhang et al., 2018). In this paper, I proposed the ideas of this paper and conducted all the experiments. Other co-authors gave me feedback at different stages.

Each human examiner then marked individual texts from different learners in a random sequence. Hence, no human examiner knew whether any pair of texts was written by the same learner or not. This removes the grading bias caused by knowing the authorship behind each text. We summarise the six datasets used in this chapter in Table 4.1.

Dataset	CEFR	Score Range	MEAN	STD	#P	#S	#train	#dev	#test
unshuffled datasets									
FCE-PUB	B2	0-20	13.92	2.92	31	1212	822	293	97
FCE-PRI	B2	0-20	14.51	2.18	37	2047	1447	300	300
CAE-PRI	C1	0-20	13.20	2.69	50	2088	1488	300	300
IELTS	A1-C2	0-9	5.78	0.96	58	1604	1004	300	300
shuffled datasets									
FCE-S	B2	0-20	13.72	2.41	67	6584	5984	300	300
CAE-S	C1	0-20	12.77	2.73	35	1910	1310	300	300

Table 4.1: Details of the six datasets used in this chapter. The letter S in FCE-S and CAE-S means these two datasets are the **shuffled** datasets. MEAN and STD describe the mean and standard deviation of the scores. #P means the number of prompts and #S means the number of scripts in the corresponding dataset.

4.3 Notation

We reuse the notation in Section 3.4.1 for our discussion. Each dataset consists of M tasks for each learner to answer, and there are J learners in one dataset. We assume that each learner only takes any exam once. All the datasets we described in Section 4.2 require learners to write two texts. Hence, $M = 2$ in each dataset. $t_{m,j}$ denotes the m^{th} text written by learner l_j, which answers the m^{th} task $task_m$ in a dataset. text $t_{m,j}$ can be represented as a sequence of words written by learner l_j. The individual score for text $t_{m,j}$ marked by a human examiner is $s_{m,j}$.

$TL_j = \{t_{m,j}|m = 1, ..., M\}$ denotes the set of all the texts written by l_j in a dataset. In other words, TL_j is equivalent to the script answered by learner l_j.

$TN_{m,j} = TL_j \setminus t_{m,j}$ denotes the neighbouring text set of $t_{m,j}$, which is all the texts written by learner l_j except for $t_{m,j}$. In this chapter, since each dataset only contains 2 texts per learner, the number of texts in $TN_{m,j}$ is always 1, and the only text in this set is $t_{(M-m+1),j}$, which denotes the neighbouring text of $t_{m,j}$.

$TT_m = \{t_{m,j}|j = 1, ..., J\}$ denotes the sequence of all the texts to the m^{th} task $task_m$ answered by all learners in the same exam.

4.4 Assumptions

The hypothesis we investigate in this chapter that using authorship knowledge to improve the performance of an ATS model relies upon two core assumptions.

The first assumption is that the individual scores should be correlated for each script. If there is a variable $skill_j$ which can describe the writing skill of each learner l_j, this skill $skill_j$ is approximately constant during an exam. If we believe the skill of a learner could be measured by the English exam they take, score $s_{m,j}$ for any task $task_m$ will be a sample from a distribution centred on $skill_j$ during the exam. We also assume that learners will not behave totally differently on the two tasks during the same exam session. In this case, these individual scores should be close and correlated well with their skill $skill_j$, and including this correlation knowledge into our individual-level model might be helpful in boosting the model performance.

However, the first assumption is not always correct. In some circumstances, learners will perform really well on some tasks, but fail to finish other tasks to the same quality, and they can get low scores on these tasks. An obvious reason for this is that some learners might have managed their time badly and failed to finish the second task; some might also be better prepared for the topic and genre elicited by one of the prompts.

To verify and measure this assumption, we calculate several metrics between all the texts to the first task (TT_1), and the second task (TT_2) answered by all the learners in each dataset. These metrics include the root-mean-squared error (RMSE), the Pearson correlation (ρ_{prs}) and the Spearman correlation (ρ_{spr}). The results are given in Table 4.2.

Dataset	RMSE	ρ_{prs}	ρ_{spr}
unshuffled datasets			
FCE-PUB	2.264	0.706	0.704
FCE-PRI	1.902	0.630	0.607
CAE-PRI	2.148	0.684	0.670
IELTS	0.716	0.746	0.735
shuffled datasets			
FCE-S	2.566	0.440	0.416
CAE-S	2.984	0.419	0.394

Table 4.2: Relation between TT_1 and TT_2 to check how the scores of the first and second text written by each learner are correlated.

As we can see, ρ_{prs} and ρ_{spr} are above 0.6 in the four unshuffled datasets, and above 0.4 in the two shuffled datasets, and the p-values for ρ_{prs} and ρ_{spr} of each dataset are both smaller than 10^{-10}. This verifies our first assumption to some degree that there is a correlation between the scores of TT_1 and TT_2. Whether this amount of agreement is

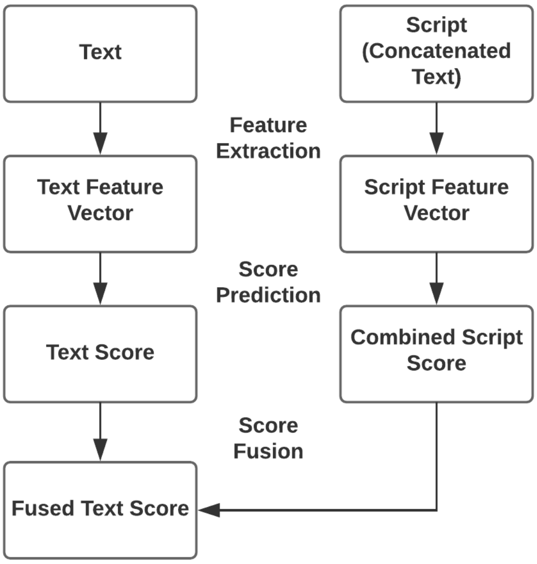


Figure 4.1: Score fusion between a text written by a learner and its corresponding script (concatenated text).

beneficial in improving the performance of an ATS model is further investigated in the following sections.

The second assumption concerns whether passing authorship knowledge to an ATS model truly improves the model performance by bringing more reliable features and better understanding about each learner's writing skill. An alternative explanation for the possible improvement, if it exists, is brought by the bias during the marking procedure. When comparing the unshuffled and shuffled datasets on RMSE in Table 4.2, FCE-S is higher than FCE-PRI, and CAE-S is higher than CAE-PRI, which suggests that human examiners might be affected when marking the second text after the first. Hence, we aim to determine whether authorship knowledge truly improves ATS performance by looking at the shuffled dataset, since any improvement on the unshuffled dataset might be the result of grading bias.

4.5 Model Fusion

In this section, we describe the approaches to pass the authorship knowledge of each text into our individual-level model described in Section 3.4. We propose two ways in which we can pass authorship knowledge into our ATS model, which we refer to as **score fusion** and **feature fusion**.

For **score fusion** (Figure 4.1), we concatenate all the texts within the same script TL_j together as ct_j. We extract the script feature vector $\mathbf{cf}_j$ from the concatenated text ct_j.

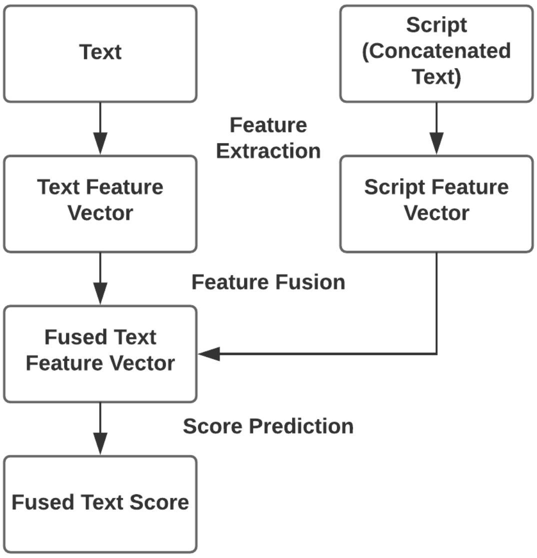


Figure 4.2: Feature fusion between a text written a learner and its corresponding script (concatenated text).

An overall-level model is trained between $\mathbf{cf}_j$ and its combined script score cs_j. This overall-level model predicts the combined script score of ct_j as $\hat{cs}_j$, and the predicted normalised combined score $\frac{\hat{cs}_j}{M}$ is fused with the predicted individual score $\hat{s}_{m,j}$ to get the predicted fused score $\hat{fs}_{m,j}$ by linear interpolation:

$$\hat{fs}_{m,j} := (1-\alpha)\hat{s}_{m,j} + \alpha\frac{\hat{cs}_j}{M}$$

The interpolation hyper-parameter $\alpha \in [0, 1]$ is tuned on the development set, and $\hat{fs}_{m,j}$ is then rounded to the nearest valid score $\hat{rs}_{m,j}$ on the given grading scale as the final predicted individual score for text $t_{m,j}$.

For **feature fusion** (Figure 4.2), we still extract the script feature vector $\mathbf{cf}_j$ from ct_j. Then, we define the fused feature vector $\mathbf{ff}_{m,j}$ of $t_{m,j}$ as the vector concatenated by $\mathbf{f}_{m,j}$ and $\mathbf{cf}_j$ together as follows:

$$\mathbf{ff}_{m,j} := (1-\beta)\mathbf{f}_{m,j} \oplus \beta\mathbf{cf}_j$$

where $\beta \in [0, 1]$ is the concatenating weight hyper-parameter to be tuned on the development set. We train an individual-level model on the fused feature vector $\mathbf{ff}_{m,j}$ and text score $s_{m,j}$, and the predicted score $\hat{s}_{m,j}$ is rounded to the nearest valid score $\hat{rs}_{m,j}$ on the given grading scale for text $t_{m,j}$. The benefit of feature fusion compared to score fusion is that this individual-level model can see the features from both the individual and overall levels at the training and test time. Hence, the model can optimise on the individual score prediction based on the features from both sides.

4.5.1 What to Fuse

Another question raised by the discussion in Section 4.5 is what to fuse. For text $t_{m,j}$ in score fusion, instead of fusing the individual score $\hat{s}_{m,j}$ with the combined script score $\hat{cs}_j$, we can also fuse $\hat{s}_{m,j}$ with the individual predicted score $\hat{s}_{(M-m+1),j}$ from the other text within the same script, which is the neighbouring text $t_{(M-m+1),j}$.

For feature fusion, when we are augmenting text feature vector $\mathbf{f}_{m,j}$ to $\mathbf{ff}_{m,j}$, we can concatenate the feature vector $\mathbf{f}_{(M-m+1),j}$ from the neighbouring text $t_{(M-m+1),j}$ instead of the script feature vectors $\mathbf{cf}_j$, which is from the concatenated text ct_j. Therefore, we have two different fusion approaches, and each approach also has two different sources to fuse.

4.5.2 Alternative View: Structural Probabilistic Models

We can put the fusion methods we proposed in this section into a broader framework named structured probabilistic models (Koller and Friedman, 2009). Without feature or score fusion, we implicitly assume score $s_{1,j}$ and $s_{2,j}$ of text $t_{1,j}$ and $t_{2,j}$ written by learner l_j are independent, while in the fusion methods we proposed, we add an explicit structural constraint so that $s_{1,j}$ and $s_{2,j}$ affect each other now. Adding structural constraints into a machine learning model is a well-known method, and this method has been proved effective in different fields such as topic modelling (Blei et al., 2003), parsing (Sha and Pereira, 2003) and skill rating (Herbrich et al., 2007).

4.6 Results and Discussion

In this section, we evaluate the baseline model in Section 3.4 and the fusion approaches proposed in Section 4.5 to investigate the influence of authorship knowledge.

The five setups we evaluate include:

BASE: baseline, which is the same SVR in Table 3.3.

SF-NT: SVR with **score fusion** between each text and its corresponding **neighbouring text**.

SF-CT: SVR with **score fusion** between each text and its corresponding **concatenated text**.

FF-NT: SVR with **feature fusion** between each text and its corresponding **neighbouring text**.

FF-CT: SVR with **feature fusion** between each text and its corresponding **concatenated text**.

setup	dev_rmse	dev_prs	dev_spr	test_rmse	test_prs	test_spr	hyper
1. FCE-PUB							
BASE	2.402	0.567	0.571	2.569	0.662	0.652	X
SF-NT	2.343+	0.611+	0.612+	2.572	0.693	0.696	0.35
SF-CT	2.318+	0.607+	0.610+	2.495	**0.696**	**0.702**	0.70
FF-NT	2.331+	0.599+	0.601+	2.529	0.688	0.688	0.30
FF-CT	**2.296+**	**0.616+**	**0.621+**	**2.460+**	0.694	0.695	0.67
2. FCE-PRI							
BASE	1.994	0.379	0.363	1.991	0.359	0.339	X
SF-NT	1.965+	**0.407**	0.386	1.979	0.371	0.347	0.18
SF-CT	1.963	0.406	0.390	**1.954+**	**0.398+**	**0.377+**	0.32
FF-NT	1.963	0.403	**0.391**	1.982	0.348	0.324	0.20
FF-CT	**1.960+**	0.405	0.390	1.974	0.354	0.333	0.25
3. CAE-PRI							
BASE	2.421	0.377	0.341	2.405	0.411	0.410	X
SF-NT	2.401	0.403	0.368	2.387	0.438	0.433	0.37
SF-CT	2.410	0.386	0.356	2.372	0.448+	0.444+	0.34
FF-NT	**2.380**	0.412	0.376	**2.356+**	**0.460+**	**0.455+**	0.50
FF-CT	2.392	**0.418**	**0.390+**	2.360	0.447	0.440	0.70
4. IELTS							
BASE	0.701	0.697	0.680	0.693	0.684	0.659	X
SF-NT	0.686	**0.725+**	**0.708+**	0.686	0.704	0.687+	0.34
SF-CT	0.690	0.715	0.701	0.691	0.689	0.667	0.33
FF-NT	**0.679+**	0.723+	0.701	0.683	0.698	0.680	0.50
FF-CT	0.685	0.716	0.697	**0.664+**	**0.720+**	**0.710+**	0.70
5. FCE-S							
BASE	1.997	0.447	0.375	2.085	0.476	0.442	X
SF-NT	1.979	0.454	0.385	2.050+	0.501+	0.463	0.23
SF-CT	1.945+	0.483+	0.407+	2.029+	0.510+	0.476+	0.33
FF-NT	1.947+	0.483+	0.391	**1.983+**	**0.541+**	**0.511+**	0.33
FF-CT	**1.917+**	**0.514+**	**0.435+**	2.017+	0.506	0.481	0.80
6. CAE-S							
BASE	2.312	0.521	0.447	2.421	0.504	0.471	X
SF-NT	2.301+	0.530+	0.457+	2.413	0.511	0.480	0.02
SF-CT	2.300	0.537	0.459	**2.346+**	**0.567+**	**0.523+**	0.78
FF-NT	2.288	0.529	0.462	2.370+	0.529	0.498	0.40
FF-CT	**2.263+**	**0.556+**	**0.484+**	2.361+	0.548+	0.513+	0.67

Table 4.3: Results of different setups. The best setup per dataset is in **bold**. Numbers in green mean improvement and red mean degradation over BASE. + means significantly better ($p < 0.05$) than BASE using the permutation randomisation test (Yeh, 2000) with 2,000 samples. No result is significantly worse than BASE.

setup	dev_rmse	dev_prs	dev_spr	test_rmse	test_prs	test_spr
SF-NT	0.028	0.028	0.027	0.046	0.028	0.028
SF-CT	0.028	0.028	0.028	0.028	0.028	0.028
FF-NT	0.028	0.028	0.028	0.028	0.046	0.046
FF-CT	0.028	0.028	0.028	0.028	0.046	0.046

Table 4.4: p-value for each approach estimated by the Wilcoxon signed-rank test (Demšar, 2006) across all the six datasets in this chapter. The metrics of each setup which are better than the baseline with $p < 0.05$ are highlighted in the green colour.

For each setup, we train an individual-level model for each exam in Table 4.1. The model of each setup is optimised on the corresponding development set of each exam using RMSE. We tune α for score fusion and β for feature fusion on each development set. We report RMSE, ρ_{prs} and ρ_{spr} in Table 4.3 for each development and each test set. The optimal α for each score fusion approach and β for each feature fusion approach are reported as α/β in Table 4.3. We can derive the following observations based on Table 4.3.

For feature fusion, feature fusion with neighbouring text (FF-NT) and concatenated text (FF-CT) is consistently better than the baseline (BASE) on all the datasets except for the FCE-PRI test set on ρ_{prs} and ρ_{spr}. For score fusion, score fusion with concatenated text (SF-CT) is better than BASE on all the six development and test sets. In contrast, score fusion with neighbouring text (SF-NT) is better than BASE on all the datasets regarding RMSE except for FCE-PUB. Both SF-CT and SF-NT are better than the baseline in terms of ρ_{prs} and ρ_{spr}. The improvement is visible on the two shuffled datasets FCE-S and CAE-S as well, and we suggest that the improvement is not merely the result of modelling grading bias.

We notice that in some cases, some methods do not perform well in terms of some metrics. For example, FF-CT and FF-NT did not get improvement on the FCE-PRI dataset in terms of ρ_{prs} and ρ_{spr}. One reason might be that we optimise our models on RMSE rather than on other metrics. There are differences between these metrics. In terms of RMSE, we can observe that SF-CT, FF-CT, FF-NT are always better than the baseline on the test sets.

We also conduct the Wilcoxon signed-rank test (Wilcoxon, 1945; Demšar, 2006) across the six datasets to see whether any setup is significantly better or worse than BASE across multiple datasets. The Wilcoxon test we use is based on the SciPy implementation[2]. As the score step magnitude of each dataset affects the Wilcoxon test for RMSE, we multiply all the scores in the IELTS dataset by two, to ensure the minimum score steps of all the datasets compared have the same value (1.0). We follow the same procedure to

[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html

conduct Wilcoxon signed-rank test in the following chapters as well. Based on the result in Table 4.4, all methods are significantly better than BASE across multiple development and test sets.

4.6.1 Fusion Hyper-parameters

We now start to discuss the hyper-parameters tuned by each approach. $\alpha, \beta > 0.5$ in each fusion approach tells the ATS model that it should favour the information from the other source over the current individual text $t_{m,j}$ being marked, and vice versa.

For the fusion with concatenated text ct_j, $\alpha > 0.5$ on FCE-PUB and CAE-S in SF-CT, and $\beta > 0.5$ for all the datasets except for FCE-PRI in FF-CT in Figure 4.3. It is still to be expected that the model favours ct_j over the current text $t_{m,j}$, because ct_j also contains $t_{m,j}$, and the information from $t_{m,j}$ is still dominant in the model even if $\alpha, \beta > 0.5$.

In comparison, we expect that the model fused with neighbouring text achieves the best performance on each dataset when α or β is smaller than 0.5, as the model should focus on the text $t_{m,j}$ being marked. The exception happens on FF-NT for IELTS and CAE-PRI, where the optimal $\beta = 0.5$.

Furthermore, Figure 4.4 visualizes the relation between RMSE and β in FF-NT. If we tune our model on the test sets, the optimal β for the CAE-PRI and IELTS datasets are even bigger than 0.5. The results of FF-NT suggest that some features from the tasks written by the same learner could be highly similar and shared in an ATS model in some cases.

When we are determining whether we should fuse each text with its corresponding concatenated or neighbouring text, there is no definite answer. For example, in terms of RMSE, FF-NT is better than FF-CT on CAE-PRI, IELTS and FCE-S, but not on the FCE-PUB, FCE-PRI and CAE-PRI test sets.

4.6.2 Score Difference

Although positive effects are observed in most cases, no method is significantly better than BASE on every dataset and metric we used. One reason might be that it is not ideal to aggregate the two texts written by the same learner together if the performance difference between these two texts is big. For example, some learners might perform well on the first task, but fail to complete the second task. This is what we have discussed in the first assumption in Section 4.4, and this assumption might be invalid in some cases. So, we conduct another study to see how the score difference between the two texts in each script affects the model performance.

We define the script score difference sd_j as the score difference between two texts $t_{1,j}$ and $t_{2,j}$ within the same script TL_j, which denotes $sd_j := |s_{1,j} - s_{2,j}|$.

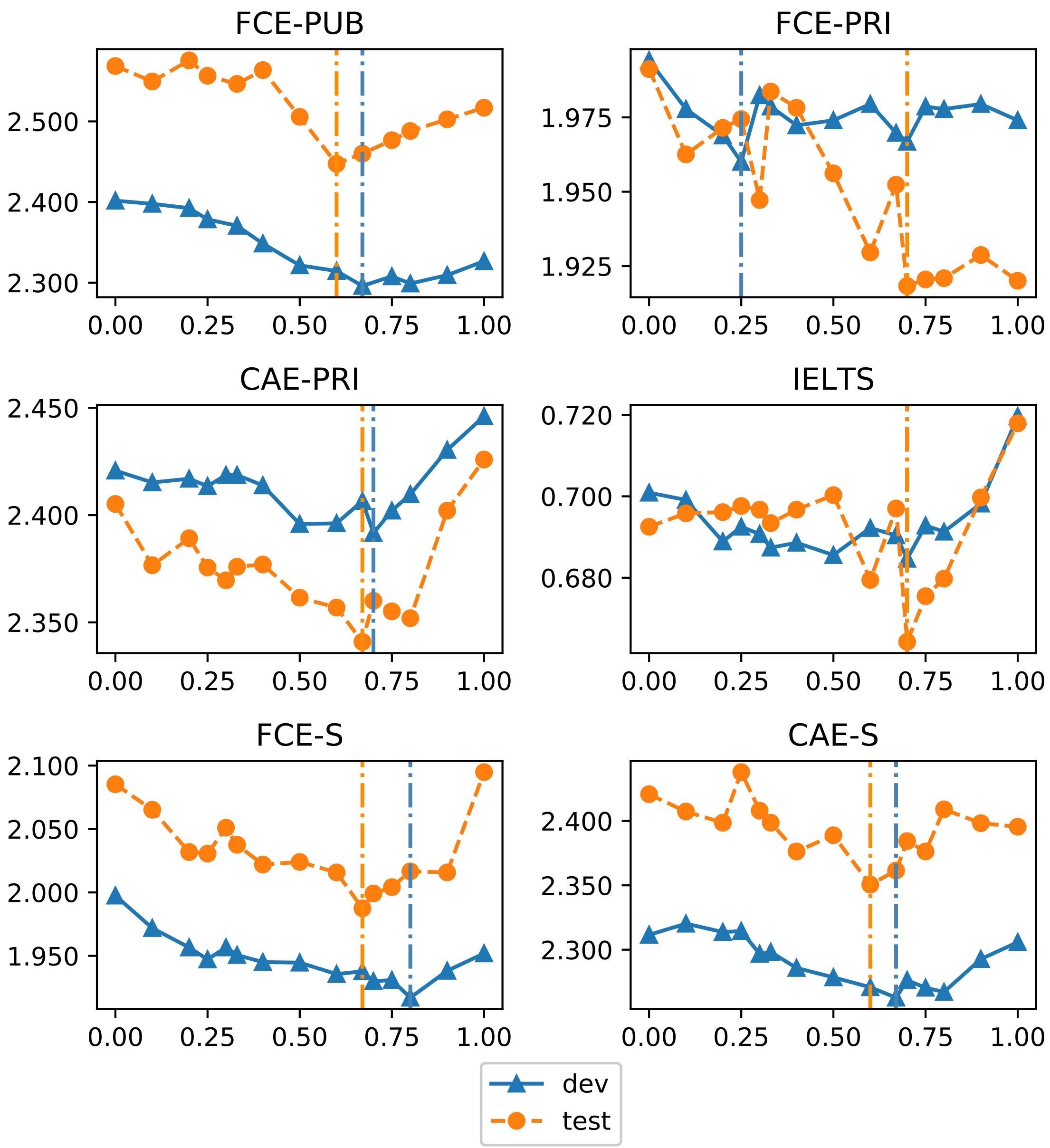


Figure 4.3: How RMSE changes with β in FF-CT. The vertical blue and orange dash-dotted lines in each graph represent that the model achieves the lowest RMSE on the development and test sets at the corresponding β.

The text score difference of $t_{m,j}$ denotes $sd_{m,j} := sd_j$.

The text score error $error_{m,j}$ denotes the difference between the predicted score and the gold score of $t_{m,j}$, and $error_{m,j} := |\hat{rs}_{m,j} - s_{m,j}|$.

The text score errors $error_{m,j}$ produced by BASE and any fusion approach on text $t_{m,j}$ denote $error^{\text{BASE}}_{m,j}$ and $error^{\text{FUSION}}_{m,j}$ respectively.

The performance difference $PD_{m,j}$ between BASE and any fusion approach for text $t_{m,j}$ denotes the difference between the errors made by two setups:

$$PD_{m,j} := error^{\text{BASE}}_{m,j} - error^{\text{FUSION}}_{m,j} \tag{4.1}$$

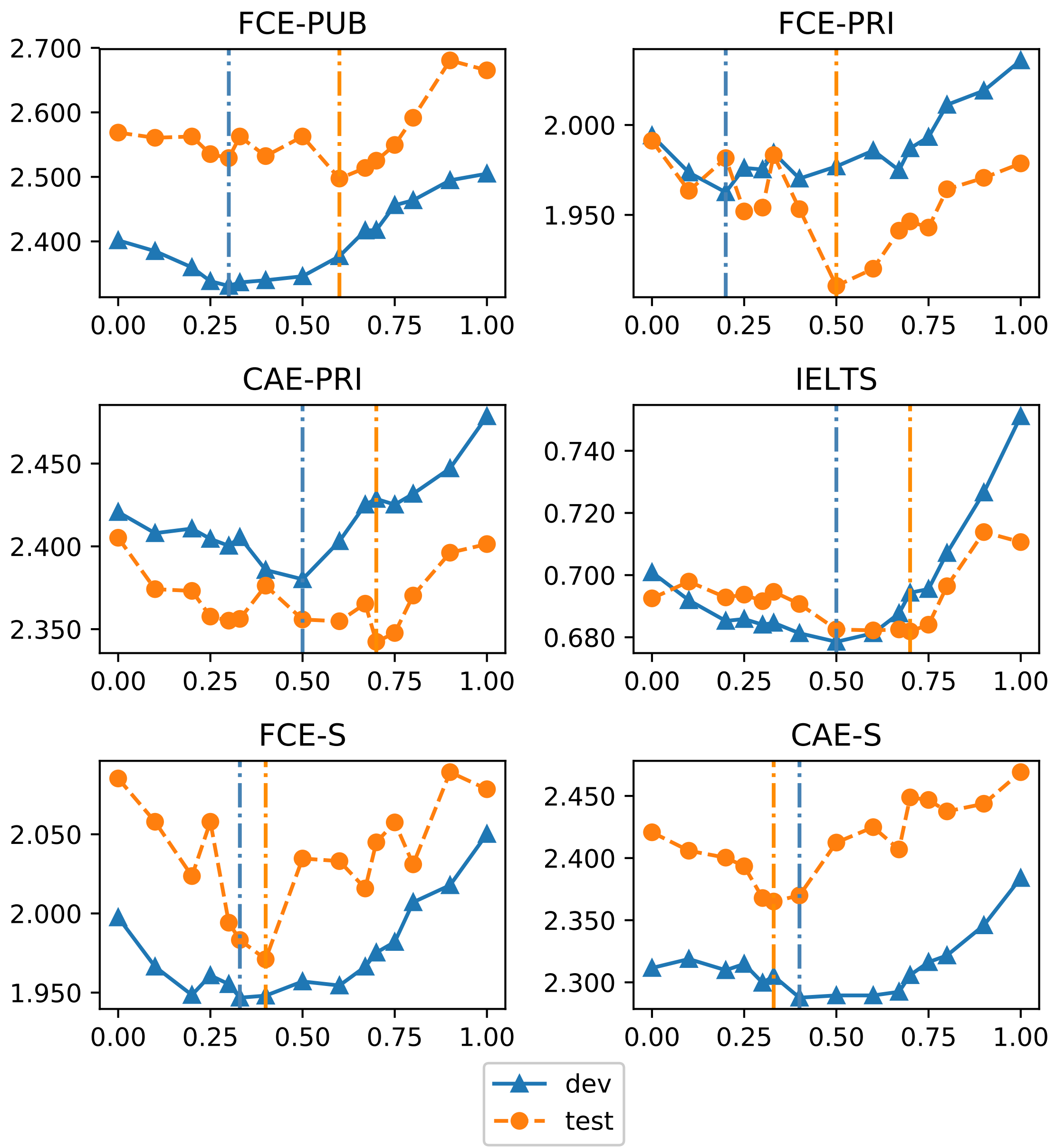


Figure 4.4: How RMSE changes with β in FF-NT. The vertical blue and orange dash-dotted lines in each graph represent that the model achieves the lowest RMSE on the development and test sets at the corresponding β.

$PD_{m,j} > 0$ means that the fusion approach is better than BASE at predicting the score of $t_{m,j}$, and vice versa.

We calculate the Pearson correlation ρ_{prs} between $PD_{m,j}$ and $sd_{m,j}$ for each development set in Table 4.5 and each test set in Table 4.6. On the one hand, most values are negative, and the few positive values in black tend to be close to 0. We suggest that there is a negative correlation between performance difference $PD_{m,j}$ and text score difference $sd_{m,j}$ on some datasets.

On the other hand, only the p-values for some negative values are smaller than

0.05. We think the score differences do not have much negative contribution on model difference, because the score differences for most scripts are relatively small (Table 4.2). This might reduce the negative influence of score difference here.

setup	FCE-PUB	FCE-PRI	CAE-PRI	IELTS	FCE-S	CAE-S
SF-NT	-0.054	-0.083*	-0.043	-0.164*	-0.123*	-0.001
SF-CT	-0.038	-0.066	0.040	-0.133*	-0.108*	-0.066
FF-NT	-0.033	-0.043	0.020	-0.035	-0.103*	0.060
FF-CT	-0.052	-0.098*	-0.030	-0.056	-0.050	-0.003

Table 4.5: Pearson correlation between performance difference $PD_{m,j}$ and script score difference $sd_{m,j}$ on the development sets. * denotes p-value < 0.05, and blue denotes a negative correlation.

setup	FCE-PUB	FCE-PRI	CAE-PRI	IELTS	FCE-S	CAE-S
SF-NT	-0.102	-0.034	-0.060	-0.188*	-0.039	-0.074
SF-CT	-0.156*	-0.009	-0.018	-0.107*	-0.039	-0.102*
FF-NT	0.002	0.036	0.034	-0.021	0.014	-0.032
FF-CT	-0.162*	0.012	-0.005	-0.108*	-0.074	-0.048

Table 4.6: Pearson correlation between performance difference $PD_{m,j}$ and script score difference $sd_{m,j}$ on the test sets. * denotes p-value < 0.05, and blue denotes a negative correlation.

4.7 Analysis of Text Length

We can view the methods utilising authorship knowledge in a way that these methods expand the length of a text, which can effectively expose more salient features for each text for our ATS model to learn. Based on the results in Table 4.3, injecting authorship knowledge to the datasets used in this chapter improves model performance in most cases. One question now is whether there are some factors behind using authorship knowledge can affect model performance, and one of the factors we will explore in this chapter is text length. More specifically, we want to study how model performance is affected if we have fewer words for each text.

To study the influence of text length in different fusion methods, for each dataset, we slice each text into two halves and treat each halve as a pseudo text. We then assign the original individual-level text score as the score of each new halved text. Therefore, the sizes of the training, development and test sets are doubled. Now, for each sliced dataset, we can train an SVR model without or with combining different fusion methods. To utilise authorship knowledge for each halved dataset, we treat the

setup	test_rmse	test_prs	test_spr	α/β
1. FCE-PUB				
BASE	2.787	0.549	0.542	-
SF-NT	2.662+	0.647+	**0.642+**	0.47
SF-CT	**2.585+**	**0.652+**	0.628+	0.81
FF-NT	2.648+	0.615+	0.595	0.70
FF-CT	2.643+	0.617+	0.601	0.67
2. FCE-PRI				
BASE	2.037	0.300	0.285	-
SF-NT	2.009	0.340	0.312	0.50
SF-CT	**1.977+**	0.343	0.318	0.73
FF-NT	1.991+	0.347+	**0.324**	0.67
FF-CT	1.979+	**0.356+**	0.321	0.67
3. CAE-PRI				
BASE	2.426	0.389	0.380	-
SF-NT	2.413	0.407+	0.399+	0.11
SF-CT	2.404	0.420	0.408	0.63
FF-NT	2.394	**0.425+**	**0.414**	0.40
FF-CT	**2.390+**	0.419	0.413	0.67
4. IELTS				
BASE	0.741	0.629	0.602	-
SF-NT	0.714+	0.670+	0.642+	0.47
SF-CT	0.704+	0.681+	0.650+	0.42
FF-NT	0.698+	0.679+	0.654+	0.50
FF-CT	**0.697+**	**0.684+**	**0.666+**	0.60
5. FCE-S				
BASE	2.129	0.428	0.417	-
SF-NT	**2.057+**	**0.495+**	0.474+	0.50
SF-CT	2.073+	0.484+	**0.477+**	0.68
FF-NT	2.074+	0.474+	0.443	0.50
FF-CT	2.080	0.470	0.453	1.00
6. CAE-S				
BASE	2.494	0.441	0.413	-
SF-NT	2.452+	0.490+	0.460+	0.44
SF-CT	2.411+	**0.524+**	**0.488+**	0.66
FF-NT	2.456+	0.477+	0.451+	0.50
FF-CT	**2.390+**	0.522+	0.480+	0.75

Table 4.7: Results for each halved dataset without/with authorship knowledge added. The numbers highlighted are the best numbers in each dataset.

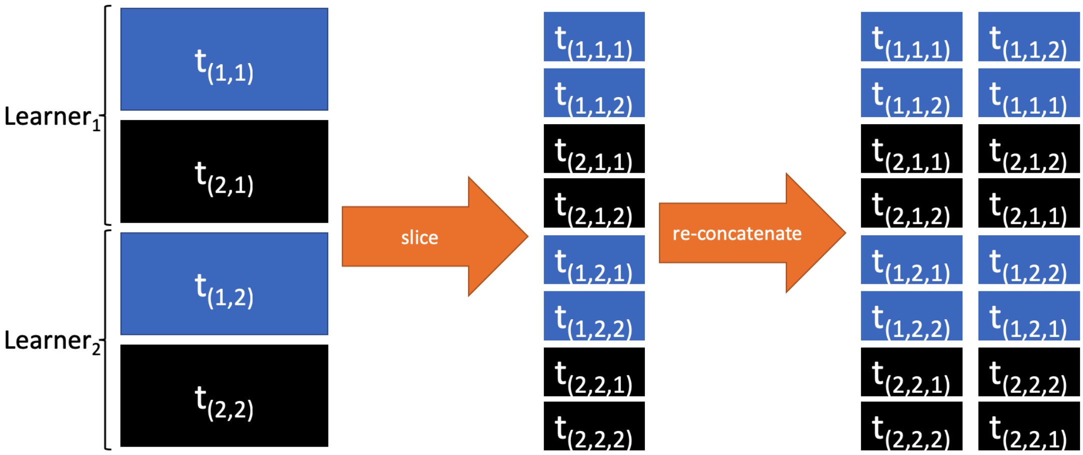

Learner$_1$
Learner$_2$
$t_{(1,1)}$
$t_{(2,1)}$
$t_{(1,2)}$
$t_{(2,2)}$
slice
$t_{(1,1,1)}$
$t_{(1,1,2)}$
$t_{(2,1,1)}$
$t_{(2,1,2)}$
$t_{(1,2,1)}$
$t_{(1,2,2)}$
$t_{(2,2,1)}$
$t_{(2,2,2)}$
re-concatenate
$t_{(1,1,1)}$ $t_{(1,1,2)}$
$t_{(1,1,2)}$ $t_{(1,1,1)}$
$t_{(2,1,1)}$ $t_{(2,1,2)}$
$t_{(2,1,2)}$ $t_{(2,1,1)}$
$t_{(1,2,1)}$ $t_{(1,2,2)}$
$t_{(1,2,2)}$ $t_{(1,2,1)}$
$t_{(2,2,1)}$ $t_{(2,2,2)}$
$t_{(2,2,2)}$ $t_{(2,2,1)}$

Figure 4.5: Illustration of halving a dataset and applying FF-NT to the halved dataset. For the mth text written by the jth learner, $t_{(m,j)}$ is sliced into two halves $t_{(m,j,1)}$ and $t_{(m,j,2)}$. To apply FF-NT to each halved text, $t_{(m,j,1)}$ is concatenated with $t_{(m,j,2)}$, and $t_{(m,j,2)}$ is concatenated with $t_{(m,j,1)}$. We then train an SVR model on the re-concatenated texts. The blue and the black rectangles in the figure refer to the texts answering the first and second writing tasks, respectively.

two halved texts originally coming from the same individual-level text as two different responses belonging to the same exam script, and we apply the same fusion methods described in Section 4.5.1 for all the halved scripts in each dataset. An example of applying FF-NT to a halved dataset is given in Figure 4.5, and the other fusion methods used in this chapter follow a similar manner.

We summarise the results of the baseline and different fusion methods applied to each halved test set in Table 4.7. In this table, we can see that all fusion methods outperform the baseline, which is different from Table 4.3 that some fusion methods perform slightly worse than the baseline in a few cases. The results might suggest that when we do not have enough words for each text, utilising authorship knowledge and expanding each text can improve model performance more effectively, compared to the circumstance when we have more words for each text.

4.7.1 Bringing Features from Different Responses

Another question related to text length is whether the improvement we observed in this chapter comes from simply writing more words or not. The background behind this question is that some exams have multiple writing tasks, and these writing tasks examine each learner from multiple aspects. Some aspects could be the same, while the others could be different between different writing tasks. Hence, for each text, if we can utilise the authorship knowledge of each text and bring more features to a text, it is worth asking which features are more beneficial in improving model performance.

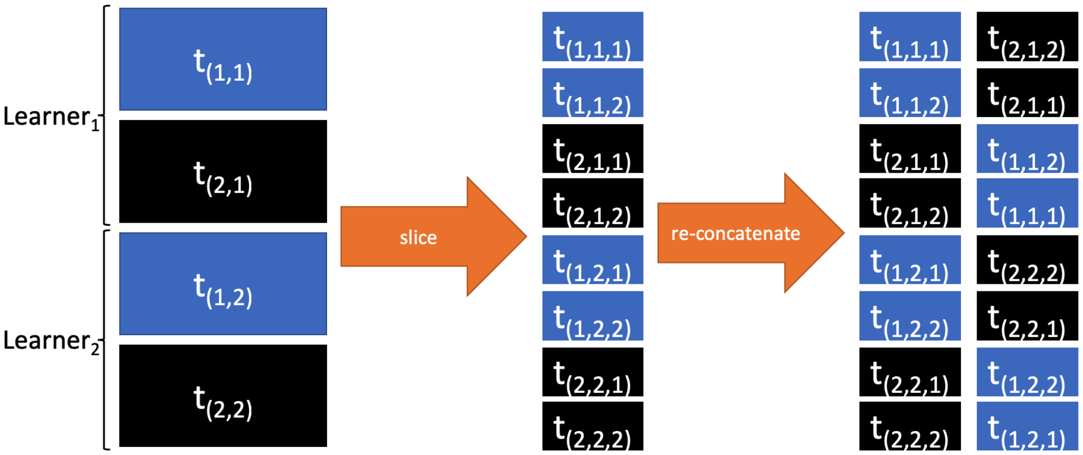

Learner$_1$
Learner$_2$
$t_{(1,1)}$
$t_{(2,1)}$
$t_{(1,2)}$
$t_{(2,2)}$
slice
$t_{(1,1,1)}$
$t_{(1,1,2)}$
$t_{(2,1,1)}$
$t_{(2,1,2)}$
$t_{(1,2,1)}$
$t_{(1,2,2)}$
$t_{(2,2,1)}$
$t_{(2,2,2)}$
re-concatenate
$t_{(1,1,1)}$ $t_{(2,1,2)}$
$t_{(1,1,2)}$ $t_{(2,1,1)}$
$t_{(2,1,1)}$ $t_{(1,1,2)}$
$t_{(2,1,2)}$ $t_{(1,1,1)}$
$t_{(1,2,1)}$ $t_{(2,2,2)}$
$t_{(1,2,2)}$ $t_{(2,2,1)}$
$t_{(2,2,1)}$ $t_{(1,2,2)}$
$t_{(2,2,2)}$ $t_{(1,2,1)}$

Figure 4.6: Illustration of halving a dataset and applying FF-NT to the halved dataset via cross concatenation. Different from Figure 4.5, during re-concatenation for halved text $t_{(m,j,k)}$, we cross concatenate $t_{(M-m+1,j,K-k+1)}$ instead of $t_{(m,j,K-k+1)}$, where $M = 2$ and $K = 2$ in this chapter.

In other words, if we have already collected some texts written by some learners, and we want to utilise authorship knowledge to improve model performance, the question here is in other to understand each learner's writing skills more accurately, whether we should simply ask learners to write more words to the **same** writing task they have already answered or to a **different** and new writing task. This simple question does not have a simple answer. Compared to asking learners to answer an extra task, the benefit of writing more words answering to the same task is that the extra words to the same task is highly relevant to the current writing task, which directly brings more relevant features to the score prediction for the current writing. In contrast, the advantage of using words to a different task is that it can diversify the features available for each text to understand a learner's writing skills more thoroughly. Therefore, in this section, we will study this question.

To study the effect of writing tasks, similar to the re-concatenation method in Figure 4.5, we still cut each text into two halves. Different from Figure 4.5, we cross concatenate halved texts written to different tasks instead of the same task. An example illustrating applying FF-NT to a halved dataset in a cross-concatenated way is given in Figure 4.6.

In Table 4.8 and 4.9, the results for the baseline (BASE) and non-cross-concatenated methods (cross_conc=FALSE) come from Table 4.7. Similar to non-cross-concatenated methods, cross-concatenated methods (cross_conc=TRUE) are always better than the baseline, which means that concatenating texts from different questions is also helpful. In terms of the best performing method for each test set, FF-CT with cross-concatenated method has achieved the best result across all the test sets except for the CAE-S test set. It suggests that feature diversity is also a useful factor in improving model performance,

setup	cross_conc	test_rmse	test_prs	test_spr
1. FCE-PUB				
BASE	-	2.787	0.549	0.542
SF-NT	FALSE	2.662+	0.647+	0.642+
SF-CT	FALSE	2.585+	0.652+	0.628+
FF-NT	FALSE	2.648+	0.615+	0.595
FF-CT	FALSE	2.643+	0.617+	0.601
SF-NT	TRUE	2.743	0.602+	0.584
SF-CT	TRUE	2.624+	0.637+	0.640+
FF-NT	TRUE	2.679+	0.598+	0.585+
FF-CT	TRUE	**2.556+**	**0.661+**	**0.657+**
2. FCE-PRI				
BASE	-	2.037	0.300	0.285
SF-NT	FALSE	2.009	0.340	0.312
SF-CT	FALSE	1.977+	0.343	0.318
FF-NT	FALSE	1.991+	0.347+	0.324
FF-CT	FALSE	1.979+	0.356+	0.321
SF-NT	TRUE	2.014	0.328	0.299
SF-CT	TRUE	1.969+	0.350+	0.331
FF-NT	TRUE	1.996+	0.347+	0.319
FF-CT	TRUE	**1.961+**	**0.362+**	**0.333+**
3. CAE-PRI				
BASE	-	2.426	0.389	0.380
SF-NT	FALSE	2.413	0.407+	0.399+
SF-CT	FALSE	2.404	0.420	0.408
FF-NT	FALSE	2.394	0.425+	0.414
FF-CT	FALSE	2.390+	0.419	0.413
SF-NT	TRUE	2.425	0.392	0.386
SF-CT	TRUE	2.404	0.414	0.407
FF-NT	TRUE	2.400	0.415	0.409
FF-CT	TRUE	**2.372+**	**0.437+**	**0.440+**

Table 4.8: Comparison between cross-concatenated and non-cross concatenated methods for the FCE-PUB, FCE-PRI and CAE-PRI halved test sets. The numbers highlighted are the best numbers in each test set.

setup	cross_conc	test_rmse	test_prs	test_spr
4. IELTS				
BASE	-	0.741	0.629	0.602
SF-NT	FALSE	0.714+	0.670+	0.642+
SF-CT	FALSE	0.704+	0.681+	0.650+
FF-NT	FALSE	0.698+	0.679+	0.654+
FF-CT	FALSE	0.697+	0.684+	0.666+
SF-NT	TRUE	0.727+	0.651+	0.623+
SF-CT	TRUE	0.706+	0.680+	0.655+
FF-NT	TRUE	0.709+	0.666+	0.646+
FF-CT	TRUE	**0.688+**	**0.691+**	**0.670+**
5. FCE-S				
BASE	-	2.129	0.428	0.417
SF-NT	FALSE	2.057+	0.495+	0.474+
SF-CT	FALSE	2.073+	0.484+	0.477+
FF-NT	FALSE	2.074+	0.474+	0.443
FF-CT	FALSE	2.080	0.470	0.453
SF-NT	TRUE	2.101	0.459	0.434
SF-CT	TRUE	2.063+	0.487+	0.475+
FF-NT	TRUE	2.090+	0.456	0.447
FF-CT	TRUE	**2.048+**	**0.509+**	**0.489+**
6. CAE-S				
BASE	-	2.494	0.441	0.413
SF-NT	FALSE	2.452+	0.490+	0.460+
SF-CT	FALSE	2.411+	**0.524+**	**0.488+**
FF-NT	FALSE	2.456+	0.477+	0.451+
FF-CT	FALSE	**2.390+**	0.522+	0.480+
SF-NT	TRUE	2.482	0.464	0.435
SF-CT	TRUE	2.443+	0.485+	0.452+
FF-NT	TRUE	2.453+	0.471+	0.442
FF-CT	TRUE	2.417+	0.494+	0.457+

Table 4.9: Comparison between cross-concatenated and non-cross concatenated methods for the IELTS, FCE-S and CAE-S halved test sets. The numbers highlighted are the best numbers in each test set.

and the improvement we observed in this chapter here might be a combination of the extra information brought by writing more words and feature diversity. Furthermore, the best performance brought by the non-cross-concatenated method on the CAE-S test set suggests that feature relevance might also affect model performance.

4.8 Summary

In this chapter, we investigated whether multi-source learning based on authorship knowledge, by means of score fusion and feature fusion, is a useful feature in ATS. We showed that including such information improves model performance, and this improvement is not only from modelling grading bias. In some operational settings, it might be considered impossible or unfair to use other responses to score a new response, and grading guidelines usually require texts to be marked in isolation. Nevertheless, we found a clear improvement when making use of such information, and no approach is significantly worse than the baseline on any metric and any dataset. In other words, the positive influence brought by our fusion approaches is stronger than the possible negative effect hidden discussed in Section 4.6.2 behind each approach. Finally, we investigated how text length affects scoring. We notice that there is more a consistent improvement for using fusion approaches when we have fewer words available for each text, and feature diversity brought by different writing tasks might also be a factor in the model performance improvement we observed in this chapter.

Chapter 5

Multi-Domain Scoring within the Same Exam

5.1 Introduction

In this chapter, we investigate how multi-domain learning can be conducted on different intra-exam properties.

As we have discussed in Section 2.5, the grading criteria for texts answering one question might be different for another question, which might a factor affecting the performance of an ATS model. In this chapter, we focus on the intra-exam properties that pertain to the same exam, whereas in the next chapter, we look at ATS across various exams. These properties act as attributes to put data into different domains, and a question behind these attributes and domains now is whether modelling the existence of attributes explicitly in an ATS system can improve or degrade system performance, and, if it can help to improve the ATS system performance, how we should do it. One way in which we can embed this attribute information in an ATS system is *multi-domain learning* (described in Section 2.5).

In this chapter we set out to investigate:

- identifying intra-exam properties that can be good attributes and define a good domain in ATS;
- examining whether multi-domain learning can improve ATS system performance.

The contributions of our work in this chapter can be summarised as follows:

- It is the first work to identify possible attributes other than prompt information for learners' texts in ATS. We divide the data we have into different domains for multi-domain learning based on the attribute information;

- We apply *frustratingly easy domain adaptation* (Daume III, 2007), a multi-domain learning technique described in Section 5.3, to model the similarities and differences between data from multiple domains;
- We apply frustratingly easy domain adaptation to a multi-attribute version (Joshi et al., 2012) in order to encourage texts with more attributes in common to share more features than they did the original method;
- We further extend the multi-attribute multi-domain learning model by weighting hyper-parameters from various attributes differently to better model the similarities and differences between the texts.
- We experimentally demonstrate that partitioning data into different domains by their attributes could gain a marginal ATS system performance improvement in most cases;
- We show that an ATS system modelling multiple attributes has marginally better performance to a system modelling a single attribute or no attribute in four out of five test sets;
- We show that there are more cases that weighted FEDA has better performance compared to the original FEDA, although the difference is not big.

5.2 Attributes

First, we must decide which attributes will be used to group texts into different domains for multi-domain learning. In our study, we identify the following attributes of each text based on the datasets presented in Section 3.3: *prompt*, *genre* and *task*. In this section, we give and summarise the definitions of these attributes.

The *prompt* attribute of a text is the prompt the text responds to, which includes all the information that learners are required to answer in their responses. An example is presented in Figure 5.1.

Prompts can be further classified into sub-categories according to the given requirements. An example of the requirement is *genre*, which refers to what type of text learners are required to write, such as writing a letter or a story.

In Chapter 4, we have seen that some exams require learners to complete two writing tasks. We define the *task* attribute of a text as the task (i.e. first or second) the text answers. We define this attribute because the writing tasks within some exams have slightly different requirements. For example, in IELTS, the minimum required word count for the first task is 150, but 250 for the second task. Thus, including this attribute in the model is thought to improve system performance potentially. Furthermore, in some datasets, the task attribute could also represent genre to some extent. The first task of the CAE-PRI dataset, for example, asks learners to write an essay, while the second task asks them to write a letter/email, proposal, report or review. The task attribute, in this case, is equivalent to whether the genre is an essay or not, and thus modelling the task attribute might capture some genre-specific information as well. In contrast, in the FCE-PRI dataset, the first task is to write an essay, but a small portion of prompts ask learners to write another essay in the second task. In the FCE-PUB dataset, the first task is to write a letter either in a formal or informal format, and some questions in the second task require learners to write a letter only in a casual style.

It would be possible to identify other text attributes, such as topic (e.g. whether the prompt requires learners to express their views on education, science, sports or other issues) or formality (e.g. whether the prompt asks learners to write in a formal or informal style). We leave further attribute identification to future research.

5.2.1 Annotation

After defining the attributes of each text, we then need to derive the possible values for each attribute. In this section, we describe how we determine and annotate the attribute information for each prompt in the datasets described in Section 3.3.

For the FCE-PUB and IELTS datasets, we have all the prompt information for all the learners' texts. To annotate the genre information, for the FCE-PUB dataset, we use the exact type the learners are required to write in each prompt. An annotation example is presented in Figure 5.5. In this figure, q="1" is the first task from which all learners are required to finish, while q="2", q="3", q="4" and q="5" are the optional questions in the second writing task from which learners can choose one and only one to answer. The highlighted words in Figure 5.5 are treated as the genre attributes for each prompt. For the fifth question q="5", there are two optional sub-questions for learners to write an article or a review for a book. We group all these book-related questions in q="5" into the *book* genre. After our annotation work, the first task in FCE-PUB contains only one genre *letter*, while the second task have six genres: *letter*, *composition*, *article*, *report*, *story* and *book*.

For the IELTS dataset, we define three genres: *data summarisation*, *letter* and *essay*. The example prompts from the IELTS academic exam in Figures 5.1 and 5.2 would be annotated as data summarisation and essay, respectively. While the IELTS general exam in Figure 5.3 and Figure 5.4 would be annotated as letter and essay, respectively. For the task attribute, we treat all the data summarisation (Figure 5.1) and letter questions (Figure 5.3) as the first task in the IELTS dataset, and all the essay questions from both exams (Figures 5.2 and 5.4) as the second task.

The chart below shows the number of men and women in further education in Britain in three periods and whether they were studying full-time or part-time.

Summarise the information by selecting and reporting the main features, and make comparisons where relevant.
Write at least 150 words

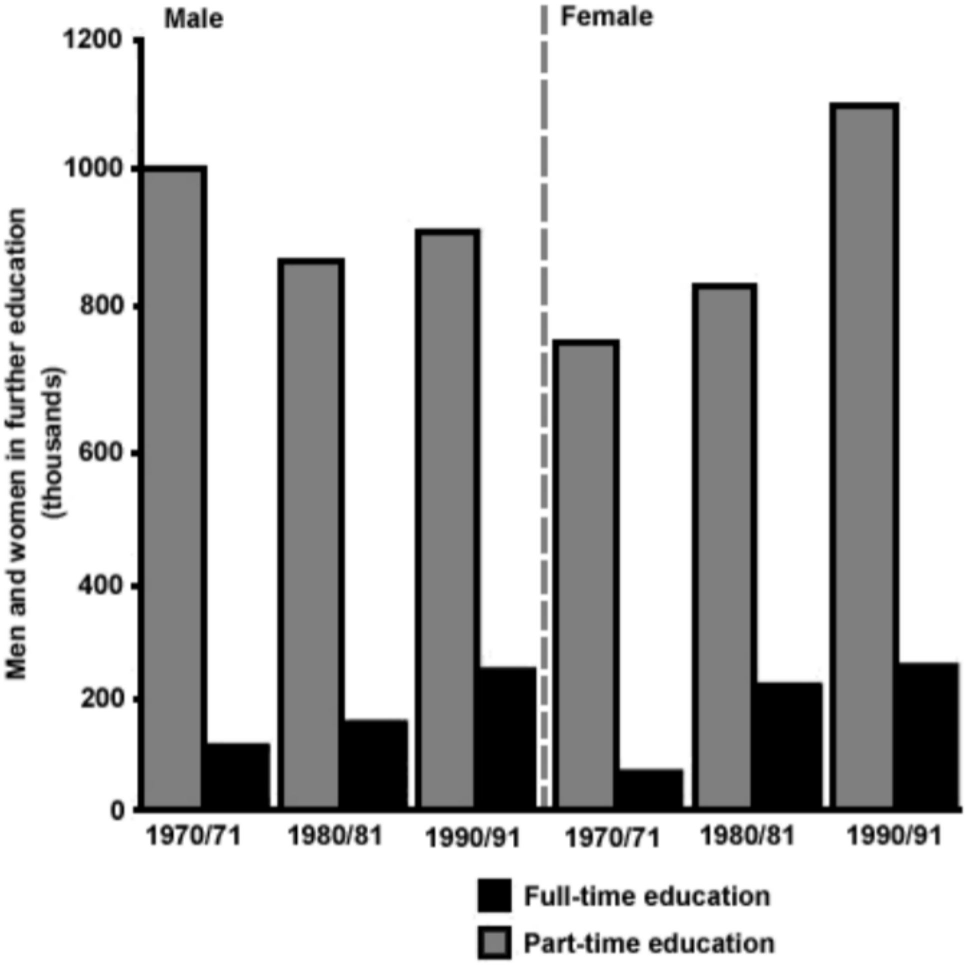


Figure 5.1: An example prompt for the first task (data summarisation) in the IELTS academic exam

The first car appeared on British roads in 1888. By the year 2000 there may be as many as 29 million vehicles on British roads.
Alternative forms of transport should be encouraged and international laws introduced to control car ownership and use.
To what extent do you agree or disagree?
Give reasons for your answer and include any relevant examples from your knowledge or experience.
Write at least 250 words.

Figure 5.2: An example prompt for the second task (essay) in the IELTS academic exam

You live in a room in college which you share with another student.
However, there are many problems with this arrangement and you find it very difficult to work.
Write a letter to the accommodation officer at the college. In the letter,

- describe the situation
- explain your problems and why it is difficult to work
- say what kind of accommodation you would prefer

Write at least 150 words.
You do NOT need to write any addresses.
Begin your letter as follows:
Dear Sir or Madam,

Figure 5.3: An example prompt for the first task (letter) in the IELTS general exam

The first car appeared on British roads in 1888. By the year 2000 there may be as many as 29 million vehicles on British roads.
Alternative forms of transport should be encouraged and international laws introduced to control car ownership and use.
To what extent do you agree or disagree?
Give reasons for your answer and include any relevant examples from your knowledge or experience.
Write at least 250 words.

Figure 5.4: An example prompt for the second task (essay) in the IELTS general exam

We do not have detailed prompt information for the KET-PRI, PET-PRI, FCE-PRI and CAE-PRI datasets, but only the prompt ID of each text answers. This prompt ID can indicate whether any two texts written answer the same or different prompts.[1] We also lack the genre information of each prompt in the FCE-PRI and CAE-PRI datasets, but we can infer the genre information of the texts from the PET-PRI and KET-PRI datasets. For PET-PRI, Question 7 asks learners to write a letter, while Question 8 asks them to write a story based on a given beginning or ending. In contrast, for the KET-PRI dataset, there is only one writing task. In this task, there is only one type of question asking learners to write a message to a friend. Hence, the only attribute information with more than one value for KET-PRI is the prompt attribute.

To annotate the task attribute of the FCE-PRI and CAE-PRI datasets, we still apply the same strategy as that used for the FCE-PUB dataset: we group all of the first questions (q="1") as the first task and all other questions as the second task.

The distribution of the prompt, genre and task attributes of each dataset is outlined in Table 5.1.

Dataset	Prompts	Genres	Tasks
KET-PRI	6	1	1
PET-PRI	16	2	1
FCE-PRI	37	-	2
CAE-PRI	50	-	2
FCE-PUB	31	6	2
IELTS	58	3	2

Table 5.1: Distribution of attributes in each dataset.

5.3 Multi-Domain Learning

Multi-domain learning is a technique used to help machine learning tasks such as ATS learn the relationship between the input and output space by discriminating data instances from miscellaneous sources. This technique helps to identify common aspects shared by data instances and discriminate between domain-specific differences, potentially improving model performance. In this section, we describe the multi-domain learning technique we use in this chapter for this purpose.

[1] The reason for this data omission is that, at the time of writing, Cambridge Assessment is still using the prompts for these texts in the exams, and the organisation wishes to ensure that the information is not leaked to the public via any channels.

```
<?xml version="1.0" encoding="UTF-8"?><xml><exam m="6" x="0100" y="2000">
Part 1
You must answer this question.
<q n="1">
You recently entered a competition and have just received this letter from the organiser.
...
Write a letter of between 120 and 180 words in an appropriate style on the opposite page.
Do not write any postal addresses.
</q>

Part 2

Write an answer to one of the questions 2 - 5 in this part.
Write your answer in 120 - 180 words in an appropriate style on the opposite page.
Put the question number in the box at the top of page 5.

<q n="2">
Your English class is going ...
Write your report.
</q>

<q n="3">
You have recently had a class discussion about shopping.
...
Write your composition.
</q>

<q n="4">
Last month, you enjoyed helping at a pop concert and your pen friend, Kim, wants to hear about your experience.
...
Write your letter.
</q>

<q n="5">
Answer one of the following two questions based on your reading of one of these set books.
Write (a) or (b) as well as the number 5 in the question box, and the title of the book next to the box.
Your answer must be about one of the books below.
<qq nn="a">
...
</qq>
Or
<qq nn="b">
....
</qq>
</q>
</exam></xml>
```

Figure 5.5: Example prompts from FCE-PUB in one exam session. The highlighted words are the genre information we annotate for each prompt. The full prompts are available in Appendix A.

5.3.1 Frustratingly Easy Domain Adaptation

Frustratingly easy domain adaptation (FEDA, Section 2.5) is a well-known multi-domain learning technique that was originally proposed by Daume III (2007). In this section, we describe this method in more details.

Assume that we have data from only two domains, D1 and D2, and the data $\{x\}$ is split as $\{x\}^{\mathrm{D1}}$ and $\{x\}^{\mathrm{D2}}$. After performing feature extraction from the data $\{x\}$ and representing each data instance x by its corresponding feature vector $\mathbf{x}$ with vector size D, FEDA defines a function $\Phi^{(aug)}$ to map $\{x\}^{\mathrm{D1}}$ and $\{x\}^{\mathrm{D2}}$ separately as:

$$\Phi^{\mathrm{aug}}(x) = \begin{cases} \mathbf{x} \oplus \mathbf{x} \oplus \mathbf{0}, & \text{if } x \in \{x\}^{\mathrm{D1}} \quad (5.1a) \\ \mathbf{x} \oplus \mathbf{0} \oplus \mathbf{x}, & \text{if } x \in \{x\}^{\mathrm{D2}} \quad (5.1b) \end{cases}$$

where $\mathbf{0} = \underbrace{\langle 0, 0, ..., 0\rangle}_{D}$ is defined as the zero vector with size D, and $\oplus$ is the vector concatenation.

The feature vectors $\mathbf{x}$ from both D1 and D2 are mapped to an augmented space X_{aug} with size $(2 + 1) \times D = 3D$. Following this, the mapped vectors are passed into the machine learning model (e.g. the Support Vector Regression model) to learn the relationship between the augmented space X_{aug} and the output space Y. In this thesis, Y represents the text scores marked by human examiners.

We can see how FEDA can handle data from different domains. Both Equation 5.1a and 5.1b contain three terms, and all data instances after feature augmentation now have domain-independent representation (the first term $\mathbf{x}$ in both equations) shared between $D1$ and $D2$, and the weight parameters from the machine learning model on this sub-space might learn the similarities shared by all the data instances from both domains. Moreover, this method preserves one domain-dependent representation (the items after the first terms $\mathbf{x}$ in Equation 5.1a and 5.1b) for each domain to learn the weights needed to model the differences between domains. If we use the second terms in both equations as an example, the zero vector $\mathbf{0}$ in Equation 5.1b cannot directly affect the weight parameters learned by the machine learning model on this sub-space, and the weight parameters of this sub-space will only directly learn from the second item $\mathbf{x}$ in Equation 5.1a to predict the output $\{y\}^{D1}$ of the data instances $\{x\}^{D1}$. The last items in both equations also proceed in a similar way that they help learn the domain-specific features for D2. Hence, the weights under this sub-space learn the unique properties of $D2$, but not $D1$.

We can view a kernelised perspective of FEDA in order to understand the influence of the data from different domains. Assume that we have a linear kernel K to encode the similarity between two vectors $\mathbf{x}$ and $\mathbf{x}'$ as:

$$K(x, x') = \langle \mathbf{x}, \mathbf{x}' \rangle \tag{5.2}$$

The kernel K^{aug} that describes the relation between x and x' after feature augmentation can be written as:

$$K^{\text{aug}}(x, x') = \begin{cases} 2K(x, x') & \text{if Domain}(x) = \text{Domain}(x') \\ K(x, x') & \text{otherwise} \end{cases} \tag{5.3}$$

We can see that a text will be twice as similar as the data instances from the same domain measured by K^{aug}, compared to the data instances from a different domain.

It is straightforward to extend this method to more than two domains using Equation 5.4. In general, to encode data from N domains, the augmented feature space X_{aug} consists of $N+1$ copies of the original feature space X. Apart from the global copy shared by all the data instances in the training set, the augmented space will have a sub-space for each domain a to encode domain-dependent information, as in Equation 5.4:

$$\Phi^{\text{aug}}(x) = \oplus_{a=0}^{N} f(x, a) \tag{5.4}$$

with $f(x, j)$ defined as follows:

$$f(x, a) = \begin{cases} \mathbf{x}, & \text{if } a = 0 \\ \mathbf{x}, & \text{if Domain}(x) = a \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{5.5}$$

5.3.2 From Attributes to Domains

We can see that a crucial question relating to FEDA and multi-domain learning is the determination of a domain. For example, in previous work in natural language processing tasks related to multi-domain learning such as sentiment analysis, the Amazon sentiment analysis dataset (Blitzer et al., 2007) gathered 8,000 product reviews associated with four types of products, and many researchers (Blitzer et al., 2007; Pan et al., 2010; Glorot et al., 2011) working on this dataset treated all reviews about the same type of product as data from the same domain. (Therefore, there are four domains in total.) They then proposed different multi-domain learning techniques to classify the reviews as positive or negative.

It is relatively easy to define a domain when there is only one discriminative attribute (e.g. the type of product a review corresponds to, in the previous sentiment analysis example). However, in many real-world contexts, such as that of ATS, data can be characterised using multiple attributes $\mathbf{attrs} = \langle attr_1, attr_2, ...attr_A \rangle$, with A attributes available (e.g. prompt, genre and task discussed in Section 5.2). Determining which attributes contribute to the domains that will be characterised in multi-domain learning is not necessarily straightforward. For example, if we were to have two texts answering two different prompts from the same genre, should these texts belong to different domains or the same domain? On the one hand, we could say that these two texts come from the same domain, but the texts may have sufficient differences relating to the key information that is meant to be included in the response. On the other hand, we could say that they come from different domains, but they still have something in common, as the genre required in both prompts is identical, and the texts might need to be similarly organised in structure when we are training the ATS model, due to genre-related constraints. Finally, it is relevant to also ask when multiple attributes are present (e.g. prompt and genre), whether we should consider only one attribute or both, in order to determine domains for FEDA.

5.3.3 Multi-Attribute Multi-Domain Learning

Joshi et al. (2013) have addressed the fact that items in a dataset can be categorised according to multiple attributes and offered suggestions as to how attributes should be selected to form domains in multi-domain learning. In their work, the authors extended several multi-domain learning techniques from single attribute to multiple attributes version.

One possible method (Equation 5.6) they suggested is treating every set of unique attributes as a single domain, such that, if there were two instances x_i and x_j with A attributes $\mathbf{attrs}_i = \langle attr_{1,i}, attr_{2,i}, ..., attr_{A,i} \rangle$ and $\mathbf{attrs}_j = \langle attr_{1,j}, attr_{2,j}, ..., attr_{A,j} \rangle$, respectively, then x_i and x_j would be from the same domain only if all attributes were identical:

$$\begin{gathered}\text{Domain}(x_i) = \text{Domain}(x_j) \iff \\ attr_{a,i} = attr_{a,j} \; \forall a \in 1, 2, ..., A\end{gathered} \tag{5.6}$$

However, this method is impractical. As Joshi et al. pointed out, the number of possible domains grows exponentially with the number of training instances, making it infeasible to train a model. In addition, another problem we notice is that sparsity might occur in the domains. Some domains have only one training instance or just a few, so this might make it impossible to train a robust ATS system or just degrade performance. This is a relatively common problem in our datasets, because the prompts we have are not evenly distributed across the texts in some datasets. We sorted the prompts in each dataset according to the number of answers given by learners and visualised the prompt frequency distribution using bar charts in Figure 5.6. The bar charts show that the prompts are not evenly distributed in each dataset, except for the KET-PRI dataset (the top left graph in Figure 5.6). Furthermore, this problem becomes even more severe in the FCE-PUB, PET-PRI, FCE-PRI and CAE-PRI datasets, as these datasets allow learners to choose one question from a group of optional questions to answer. Some questions in these datasets are rarely selected and answered by learners, because they are hard to answer, or learners prefer answering other questions. In consequence, this reduces the number of texts responding to certain prompts available for training our model.

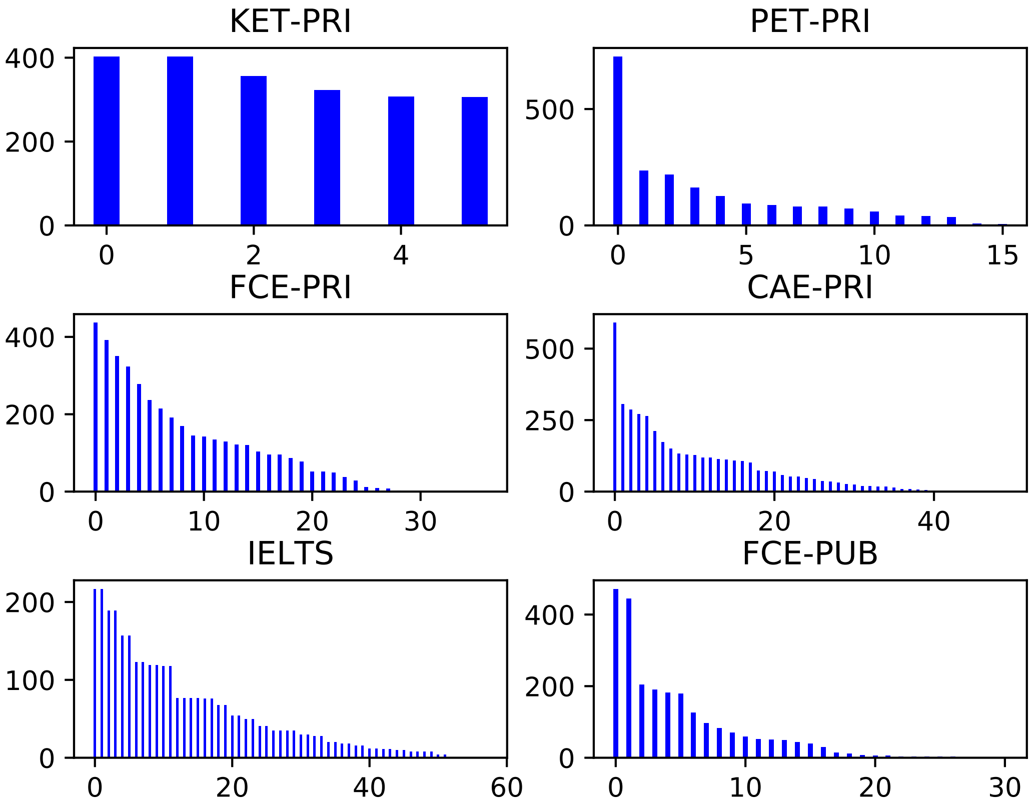
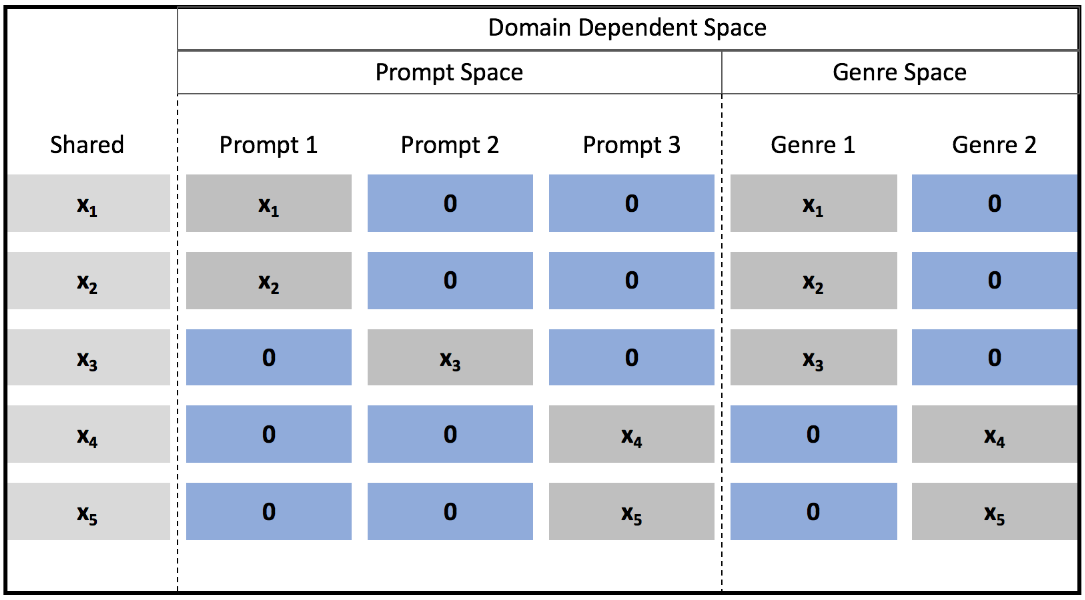


Figure 5.6: Prompt distribution in each dataset.

To avoid problems of this nature, Joshi et al. proposed another method to extend multi-domain learning techniques, including multi-domain learning by confidence-weighted parameter combination (Dredze et al., 2008, 2010) (described in Section 2.5) and FEDA, into multiple attribute versions. For multi-attribute FEDA, each unique attribute value is given a parameter sub-space, and the original feature vector $\mathbf{x}$ is mapped from $\mathbb{R}^D$ into a larger space $\mathbb{R}^{D(1+\sum_m K_m)}$, where K_m represents the number of possible values for the m-th attribute. This representation still preserves a shared space for all possible attribute values, and each attribute value has an independent parameter sub-space for learning the features of the specific attribute value. During feature augmentation, each data instance copies its features from the original space into all sub-spaces in which the data instance belongs. An example is provided in Figure 5.7. In this figure, there are five data instances with two attributes: prompt and genre. There are three distinct values (ranging from Prompt 1 to Prompt 3) for the prompt attribute and two values available for the genre attribute. Prompt 1 and Prompt 2 are from Genre 1, and Prompt 3 is from Genre 2. Data instances x_1 and x_2 come from Prompt 1; x_3 comes from Prompt 2, and x_4 and x_5 come from Prompt 3. Therefore, the augmented space of the multi-attribute FEDA is $\mathbb{R}^{(1+2+3)D} = \mathbb{R}^{6D}$. All data instances copy their original features from the shared space to the corresponding sub-spaces (Prompt Space and Genre Space in Figure 5.7), leaving the other sub-spaces as $\mathbf{0}$.



	Domain Dependent Space				
	Prompt Space			Genre Space	
Shared	Prompt 1	Prompt 2	Prompt 3	Genre 1	Genre 2
$\mathbf{x_1}$	$\mathbf{x_1}$	**0**	**0**	$\mathbf{x_1}$	**0**
$\mathbf{x_2}$	$\mathbf{x_2}$	**0**	**0**	$\mathbf{x_2}$	**0**
$\mathbf{x_3}$	**0**	$\mathbf{x_3}$	**0**	$\mathbf{x_3}$	**0**
$\mathbf{x_4}$	**0**	**0**	$\mathbf{x_4}$	**0**	$\mathbf{x_4}$
$\mathbf{x_5}$	**0**	**0**	$\mathbf{x_5}$	**0**	$\mathbf{x_5}$

Figure 5.7: Illustration of multiple attribute FEDA.

From the perspective of the kernelised space, if data instances x and x', with feature vectors $\mathbf{x}$ and $\mathbf{x}'$, were to have three attributes in common with similarity $K(x, x')$, the similarity calculated by kernel K^{aug} between $\mathbf{x_i}$ and $\mathbf{x_j}$ then would be $(3 + 1) \times K^{aug}(x, x') = 4K(x, x')$. The method simply forces data instances with more attributes in common to have greater similarity in the augmented space.

5.3.4 Weighted FEDA

As demonstrated in Equation 5.3, one problem with FEDA and multi-attribute FEDA in the kernelised space perspective is that if two data instances writing to the same prompt have n domains in common, they have n times more influence on each other than they do with the data instances with no domain in common. This implicitly assumes that the influence of all domains are identical to their domain-independent representations.

However, this might not be accurate. Given that prompts from the same exam, for example, these prompts might have many commonalities in terms of the grading criteria, and the n times more influence for the texts from the same prompt might be too high. Therefore, we suggest that a more appropriate way of conducting multi-domain learning is to assign different weights $\mathbf{aw} = \langle aw_1, aw_2, ..., aw_A \rangle$ for A attribute values:

$$\Phi^{\text{aug}}(x) = \oplus_{a=0}^{A} f(x,a) \times aw_a \tag{5.7}$$

These weights, **aw**, are the hyper-parameters to be tuned on the development set.

To reduce the **aw** search space, all aw from the same attribute type are fixed to the same value for any two different attribute values $a, a' \in A$:

$$\begin{aligned} aw_a &= aw_{a'} \\ \text{if Type}(a) &= \text{Type}(a') \end{aligned} \tag{5.8}$$

where Type could be prompt, genre or task. We have three unique weights, and each weight corresponds to one type of attribute in this chapter.

5.3.5 Relation between W-FEDA and Other Types of Multi-Domain Learning Methods

In this section, we give an analysis of the relation between W-FEDA and other types of multi-domain learning methods.

Lu et al. (2016) proposed a general framework to describe some domain adaptation and multi-domain learning methods by adding regularisation terms to force the model parameters for similar domains to be close. They derived this framework as Theorem 2.3 in their paper to summarise the objective functions minimised in the methods proposed by Evgeniou and Pontil (2004); Daume III (2007); Finkel and Manning (2009):

$$\begin{aligned} &\min_{\mathbf{w}_1,\mathbf{w}_2,\ldots,\mathbf{w}_N,\mathbf{w}_0} [\sum_{i=1}^{N} \mathcal{L}_i'\left(\mathbf{D}^i; \mathbf{w}_i, \mathbf{w}_0\right) + \left(\lambda_0 \|\mathbf{w}_0\|^2 + \sum_{i=1}^{N} \lambda_i \|\mathbf{w}_i\|^2\right)] \\ = &\min_{\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_N} [\sum_{i=1}^{N} \mathcal{L}_i\left(\mathbf{D}^i; \mathbf{v}_i\right) + \left(\sum_{i=1}^{N} \eta_{0,i} \|\mathbf{v}_i\|^2 + \sum_{1\le j<k\le N} \eta_{j,k} \left\|\mathbf{v}_j - \mathbf{v}_k\right\|^2\right)] \end{aligned} \tag{5.9}$$

In Equation 5.9 (Theorem 2.3 in their paper), we have N domains categorised by different attributes.

- In the left-hand side (LHS) of Equation 5.9, $\mathcal{L}_i'\left(\mathbf{D}^i; \mathbf{w}_i, \mathbf{w}_0\right)$ is the model loss on the data from domain $\mathbf{D}^i$, $\mathbf{w}_0$ is the shared-representation weights for all N domains, and $\mathbf{w}_i$ corresponds to the domain-specific representation for domain $\mathbf{D}^i$. $\lambda_i \geq 0$ is the hyper-parameter to control the regularisation on $\mathbf{w}_i$;

- In the right-hand side (RHS) of Equation 5.9, we define $\mathbf{v}_i \equiv \mathbf{w}_0 + \mathbf{w}_i$, and we can always rewrite LHS to hide $\mathbf{w}_0$ into RHS by[2]:

$$\begin{aligned} & \lambda_0 ||\mathbf{w}_0||^2 + \sum_{i=1}^{N} \lambda_i ||\mathbf{w}_i||^2 \\ = & \sum_{i=1}^{N} \eta_{0,i} ||\mathbf{w}_0 + \mathbf{w}_i||^2 + \sum_{1 \leq j < k \leq N} \eta_{j,k} ||\mathbf{w}_j - \mathbf{w}_k||^2 \\ = & \sum_{i=1}^{N} \eta_{0,i} ||\mathbf{v}_i||^2 + \sum_{1 \leq j < k \leq N} \eta_{j,k} ||\mathbf{v}_j - \mathbf{v}_k||^2 \\ & \textit{where } \eta_{i,j} = \frac{\lambda_i \lambda_j}{\sum_{l=0}^{N} \lambda_l} \end{aligned} \tag{5.10}$$

$\mathcal{L}_i\left(\mathbf{D}^i; \mathbf{v}_i\right)$ is still the loss on the data from domain $\mathbf{D}^i$, $\eta_{0,i} \geq 0$ is the hyper-parameter to control the regularisation on $\mathbf{v}_i$, and $\eta_{j,k} \geq 0$ controls the similarity between $\mathbf{v}_j$ and $\mathbf{v}_k$.

To give an example of Equation 5.9, Lu et al. concluded that FEDA proposed by Daume III (2007) is a special case when we have only two domains (Domain $\mathbf{D}^1$ and $\mathbf{D}^2$ for the source and target domains, respectively), and Equation 5.9 for Daume III's FEDA could be written as:

$$\begin{aligned} & \min_{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_0} [\mathcal{L}'_1\left(\mathbf{D}^1; \mathbf{w}_1, \mathbf{w}_0\right) + \mathcal{L}'_2\left(\mathbf{D}^2; \mathbf{w}_2, \mathbf{w}_0\right) + \lambda\left(||\mathbf{w}_1||^2 + ||\mathbf{w}_2||^2 + ||\mathbf{w}_0||^2\right)] \\ = & \min_{\mathbf{v}_1, \mathbf{v}_2} [\mathcal{L}_1\left(\mathbf{D}^1; \mathbf{v}_1\right) + \mathcal{L}_2\left(\mathbf{D}^2; \mathbf{v}_2\right) + \frac{1}{3}\lambda\left(||\mathbf{v}_1||^2 + ||\mathbf{v}_2||^2 + ||\mathbf{v}_1 - \mathbf{v}_2||^2\right)] \end{aligned} \tag{5.11}$$

To supplement what were not fully discussed in Lu et al. (2016), we can see that W-FEDA is equivalent to Equation 5.9 if we do not enforce λ_i in LHS categorised by the same type of attribute to be the same value like what we did in Equation 5.8. The multi-attribute version of FEDA (Joshi et al., 2013), which is a special case of W-FEDA, is equivalent to Equation 5.9 when:

- we force $\eta_{m,n} = \eta_{j,k}$ for any $m, n, j, k \in \mathbb{N}_1^N$ if m and n are the same type of attribute, so are j and k;

- $\eta_{m,n} = 0$ if m and n are not the same type of attribute.

Hierarchical Bayesian domain adaptation (Finkel and Manning, 2009) (Section 2.5.1)

[2] Please refer to the original paper for the proof of this lemma.

is also a special case of W-FEDA by adjusting the hyper-parameters η in RHS in Equation 5.9 by:

$$\eta_{j,k} \begin{cases} > 0 \text{ if } Parent(D^j) = D^k \\ = 0 \text{ otherwise} \end{cases} \tag{5.12}$$

Different from hierarchical Bayesian domain adaptation, W-FEDA does not require the attributes to be organised into a hierarchy.

5.4 Results

In this section, we report the results and discuss the multi-domain learning with different attributes under different setups.

5.4.1 Weighted FEDA on Single Attribute

First, we evaluate weighted FEDA (W-FEDA) on each single attribute in all the datasets (KET-PRI, PET-PRI, FCE-PRI, CAE-PRI, IELTS and FCE-PUB) described in Chapter 3.

W-FEDA is still implemented using support vector regression (SVR) based on LIBLINEAR with L2 loss, as described in Section 3.4.2. The epsilon hyper-parameter is set to the default value of 0.1. The cost hyper-parameter of SVR is tuned on the same group of hyper-parameters as the baseline model on the development set. For each dataset, all models with the lowest root-mean-square error (RMSE) on the development set are further evaluated on the test set. We report the results in Table 5.2.

In this table, there are six sub-tables, each corresponding to one of the datasets described in Chapter 3. In each sub-table, the first column ATTR describes the attribute we use to conduct multi-domain learning. The attribute NONE means that multi-domain learning is not used to discriminate any attribute. Hence, the first line in each sub-table is the baseline for each dataset. The other attribute values in the attribute column, PROMPT, GENRE and TASK indicate the attributes we use to conduct multi-domain learning. In other words, we treat the data instances with the same attribute value as the data belonging to the same domain, leaving only one domain to be discriminated. The second to fourth columns (dev_rmse to dev_spr) report the performance measures on the development set in each dataset, which include the root-mean-square error (RMSE), and Pearson's ρ_{prs} and Spearman's ρ_{spr} correlations. The fifth to eighth columns (test_rmse to test_spr) correspond to model performance on each test set.

From Table 5.2, we can derive the following observations. First, W-FEDA improves model performance in more than half of the cases, relative to the baseline model training a single grader for all texts in all domains without discerning any domain. In terms of

ATTR	dev_rmse	dev_prs	dev_spr	test_rmse	test_prs	test_spr
1. KET-PRI						
NONE	0.970	0.661	0.585	0.968	0.706	0.648
PROMPT	**0.952**	**0.647**	**0.595**	**0.904+**	**0.751+**	**0.723+**
2. PET-PRI						
NONE	2.247	0.594	**0.541**	2.179	0.567	0.467
PROMPT	2.216	0.601	0.540	**2.121**	**0.597**	**0.494**
GENRE	**2.214**	**0.611**	**0.541**	2.139	0.591	0.491
3. FCE-PRI						
NONE	1.994	0.379	0.363	1.991	0.359	0.339
PROMPT	**1.968**	0.399	0.379	**1.960**	0.369	0.350
TASK	1.969+	**0.404+**	**0.386**	1.979	**0.374**	**0.354**
4. CAE-PRI						
NONE	2.421	0.377	0.341	2.405	0.411	0.410
PROMPT	**2.381**	**0.412**	**0.394+**	2.398	0.436	0.435
TASK	2.417	0.385	0.346	**2.368+**	**0.439**	**0.438**
5. IELTS						
NONE	0.701	0.697	0.680	0.693	0.684	0.659
PROMPT	0.706	0.695	0.674	0.703	0.674	0.642
GENRE	0.700	0.698	0.684	**0.686**	**0.696**	**0.670**
TASK	**0.694**	**0.705**	**0.692**	0.689	0.693	**0.670**
6. FCE-PUB						
NONE	2.402	0.567	0.571	2.569	0.662	0.652
PROMPT	2.377	0.577	0.571	2.662-	0.660	0.655
GENRE	2.378	0.572	0.576	**2.548**	0.661	0.659
TASK	**2.373**	**0.582**	**0.579**	2.556	**0.687**	**0.677**

Table 5.2: W-FEDA on each single attribute in each dataset. The highlighted numbers are the best results in each dataset. + and - means significantly better/worse ($p < 0.05$) than the baseline (NONE) using the permutation randomisation test (Yeh, 2000) with 2,000 samples, respectively.

the performance on the development sets, the baseline only has the same best result on the Spearman correlation ρ_{spr} in PET-PRI as the W-FEDA expanded model on the genre attribute. In terms of the test sets, all best results are always produced by multi-domain learning.

As for statistical significance, W-FEDA achieves significant improvement on the development sets, including:

- all metrics on the FCE-PRI development set with the task attribute; and
- ρ_{spr} on the CAE-PRI development set with the prompt attribute;

For the test sets, W-FEDA shows significant improvement on:

- RMSE, ρ_{prs} and ρ_{spr} on the KET-PRI test set with the prompt attribute;
- RMSE on the CAE-PRI test set with the task attribute;

Let us analyse multi-domain learning on each single attribute. We start with the prompt attribute first. W-FEDA on the prompt attribute improves performance on the KET-PRI, PET-PRI, FCE-PRI and CAE-PRI datasets. However, for the IELTS and FCE-PUB test sets, there is a performance drop. Based on these performance measures, we suggest that only differentiation according to the prompt information is not always effective. In our datasets, some prompts are more similar than others, like requesting learners to write their texts in a specific genre, and FEDA on prompt information only ignores this potentially useful information. In addition, for the IELTS dataset, the number of prompts in Figure 5.6 is also the highest among the six datasets, and most prompt dependent features might not have enough texts to correctly learn the prompt specific features, which also might be a reason for performance drop on the IELTS dataset.

We also suggest another reason for the performance drop on the test set for the prompt attribute in the FCE-PUB dataset. In the original data setup of Yannakoudakis et al. (2011), the training and development sets are comprised of texts written in the year 2000, and the test set consists of texts collected from the year 2001. Hence, there is no overlap between the prompts from the training/development sets and those from the test set. The W-FEDA on the single prompt attribute treats all prompts identically and does not capture any implicit similarities shared (e.g. genre). Therefore, the model might overfit to the prompts existing in the year 2000 and poorly generalise on the 2001 data. This is another reason why multi-domain learning only on prompts may not be wise: the model might only optimise to the prompts in the training set, and not for future and unseen prompts. It is quite typical in many English exams to add new prompts in each year to meet new requirements and developments in language education.

For the genre attribute, performance improvement is shown across the development and test sets for the PET-PRI, IELTS and FCE-PUB datasets, with the exception of the Pearson correlation ρ_{prs} on the test set in the FCE-PUB dataset.

The task attribute follows a similar improvement trend to that of the genre attribute. Multi-domain learning on the task attribute achieves better results than the baseline model across the development and test sets for the FCE-PRI, CAE-PRI, IELTS and FCE-PUB datasets.

It is difficult to identify the dataset in which each attribute produces the best performance. Concerning RMSE on the test set for the attribute prompt, for example, W-FEDA achieves the best results on the KET-PRI, PET-PRI and FCE-PRI datasets, but not on the others. We will display and summarise the best results for each single attribute in a tidier way in Section 5.4.3.

5.4.2 Model Inspection

To closely examine the behaviour of our multi-domain learning models, in this section, we find some features that the weights for some domains are much higher than others. We list some features and analyse the weights learned for these features by the models.

We begin with the prompt-expanded FEDA for the KET-PRI dataset. Although we do not have the actual prompt text, we can infer the prompt for each answer based on the basis of words used in each response, the corresponding scores assigned by human examiners and the grading criteria for the KET exam. All six prompts ask learners to write a message to a friend. Below, we briefly describe the six prompts we infer:

- P1: writing a letter inviting a friend to watch a match;
- P2: describing a class the learner recently takes, why s/he takes it and when it will finish;
- P3: telling a friend about the camping site the learner is taking with his/her family, what s/he is doing there and when s/he will be back;
- P4: planning for shopping and describing the reasons why the learner wishes to make purchases;
- P5: describing the details of a cooking club the learner is going to with a friend;
- P6: describing a movie the learner is going to watch, what it is about and when it will finish.

The first pattern we found is related to the word unigram *do* and the bigram *do you*. The weights learnt for these two features are listed in Table 5.3. In this table, the

Feature	Shared	P1	P2	P3	P4	P5	P6
do you	-0.039	**0.494**	-0.134	-0.259	-0.098	0.045	-0.126
do	-0.186	**0.180**	-0.220	-0.138	-0.124	-0.076	0.007

Table 5.3: Word feature weights for the prompt-expanded model on the KET-PRI dataset. The highest values are highlighted in each row.

> Hello Ali!
> I have got tickets for a sport match on Saturday. Is playing my brother team. I want to go because I want to see the new stadium. **Do you want to come?**

Figure 5.8: Example answer to P1, receiving a score of 5 out of 5.

column Shared outlines the weight shared by all the domains, while P1 to P6 represent the weights for the six prompts we identified in the KET-PRI dataset. This table shows that *do you* and *do* generally have a negative weight (or a positive but relatively small weight) for the shared representation and the domain-dependent representations for P2 to P5, in contrast to P1. For the domain-dependent representation of P1, *do you* has a weight of 0.494, and *do* has a weight of 0.180. Both of them are at least ten times higher than the second highest weight in this table. We suggest that the reason for this is that *do you* and *do* are some beginner-level words used by learners, so the use of such words tends not to reward extra points for learners' language ability scores. In some cases, learners might even receive a penalty for overusing these simple words and phrases. In P1, these features receive relatively high weights, because P1 asks learners to write an invitation letter, and examiners might expect learners to write sentences such as *Do you want to come?* for this purpose, which is a key point for this specific prompt. An example is presented in Figure 5.8.

Shared	P1	P2	P3	P4	P5	P6
0.198	**0.748**	-0.004	0.000	-0.240	-0.030	-0.077

Table 5.4: Feature weights learned for the POS trigram TO VV0 NP1 in the KET-PRI dataset.

Another pattern we found in the prompt-expanded model relates to the weight for the POS trigram TO + infinitive verb + singular proper noun (TO VV0 NP1) in Table 5.4. The weight of this feature for P1 is still much higher than that of the shared space and the other five prompts. An example is given in Figure 5.9. Here, *to see Barca* is a key point to describe the event to go to, and our model highlights this prompt-relevant feature.

In summary, based on the observations presented in Tables 5.3 and 5.4, we suggest that performing prompt expansion can help to capture some prompt-specific patterns that are beneficial for scoring.

We then looked at the trigram missing rate estimated on the UKWaC British English corpus for the genre-expanded W-FEDA on the IELTS dataset. For each learner's text, Yannakoudakis et al. (2011) counted how many trigrams were not present in a background corpus of well-formed text, and they treated such missing trigrams as erroneous trigrams and fed the missing rate as a feature to their model. As presented in Table 5.5, the weight for data summarisation is much higher than the weights for the essay and letter genres. The reason we suggest might be that the data summarisation questions require learners to use numerical words to describe the information in the given figures. While phrases containing numerical words are the vital points to get scores, they are not commonly found in the background corpus. Therefore, the weight of the missing trigrams for the data summarisation task is relatively higher than the weights for the other two genres. To support this suggestion, we summarise the missing trigram rate fore each genre in Table 5.6. We can see that the normalised missing trigram rate (E/T/W * 100) for data summarisation (Data-S) is higher than the other two types of genre. Furthermore, 47.94% of the missing trigrams for data summarisation contain at least one numerical word whose POS tag is MC, which is much higher than essay (1.42%) and letter (7.22%).

Shared	Essay	Data Summarisation	Letter
-1.608	-0.956	**1.232**	-1.080

Table 5.5: Feature weights learned for the trigram missing rate estimated on the UKWaC British English corpus.

Finally, we examine weights learned for the task-expanded W-FEDA trained on the IELTS dataset with attention to word count in learners' texts. As mentioned in Section 5.2, the minimum word count requirements for the two tasks in the IELTS exam is different, as Task 2 requires more words than Task 1. We display the weights learned for the word count in Table 5.7. The table shows that the weight for Task 2 is higher than that of Task 1 for this feature, although the difference is not large.

In summary, based on the results and discussion presented in Sections 5.4.1 and 5.4.2, we conclude that different text attributes of the learners' texts can affect grader performance. We also conclude that explicitly modelling the existence of attributes in

> Hello Ali
> On Saturday I will go **to see Barca** vs Madrid. I wont to go there because is an interesting sports match. If you want, you can come with me because I have tickets. Bye

Figure 5.9: Example answer to P1, receiving a score of 5 out of 5.

Genre	#Miss	#Text	#Word	M/T/W * 100	#MC	#MC_Miss	MC_M/M
Essay	164319	1004	322.31	50.78	558	2341	1.42%
Letter	10952	128	206.93	41.35	185	791	7.22%
Data-S	111193	876	200.55	**63.29**	**12556**	**53308**	**47.94%**

Table 5.6: Trigram missing rate for each genre in the IELTS training set. For each genre, #Miss corresponds to the number of missing trigrams identified during feature extraction. #Text represents the number of texts. #Word represents the average length for each text. M/T/W * 100 is the normalised missing trigram rate (#Miss multiplied by 100 and divided by #Text and #Word). #MC represents the count of numerical unigrams (e.g *first*, *17*) whose POS tags are MC (cardinal number). #MC_Miss represents the number of missing trigrams which contain at least one numerical unigram. MC_M/M is the normalised percentage for missing numerical trigrams.

Shared	Task 1	Task 2
5.801	1.317	1.583

Table 5.7: Feature weights learned for the word count of a learner's text

a model can improve performance in some cases. However, the task of determining which attributes to model is not straightforward, and improvement does not appear across all the datasets, and most improvements are not significant. This motivates us to evaluate W-FEDA with multiple attributes modelled in the following section.

5.4.3 Multiple Attributes versus Best Single Attribute

In this section, we report the results of the W-FEDA expanded on multiple attributes. The results are presented in Table 5.8. The five sub-tables in this table map to the five datasets with more than one attribute available. In contrast to Table 5.2, Table 5.8 does not include the results for the KET-PRI dataset, because the KET-PRI dataset has only one available attribute. In each sub-table, each row represents a model as follows:

- *baseline* is the support vector regression baseline implemented in Chapter 3;
- *1-best* is W-FEDA on a single attribute with the best performance on the development set in terms of RMSE;
- *1-oracle* is W-FEDA on a single attribute with the best performance on the test set in terms of RMSE;
- *all* is W-FEDA on all attributes together with the best performance on the development set in terms of RMSE.

The *ATTR* column describes the attribute on which we conduct W-FEDA for 1-best and 1-oracle.

Before we compare the results of W-FEDA on multiple attributes and W-FEDA on each single attribute, let us compare the results between 1-best and 1-oracle. We can see that, with the exception of the FCE-PRI dataset, the best attributes chosen by 1-best and 1-oracle are different. This means that selecting the best single attribute on the development set to conduct multi-domain learning does not always produce the best performance on the unseen test set, as each attribute might capture different useful information.

Next, we compare W-FEDA on multiple attributes with the other three setups including baseline, 1-best and 1-oracle on the development sets. Compared to the baseline, W-FEDA on all attributes always perform better, except for ρ_{spr} on the PET-PRI dataset.

For the results of the test sets in Table 5.2, W-FEDA shows the best performance across the PET-PRI, FCE-PRI and CAE-PRI test sets. For the IELTS test sets, W-FEDA has the best performance in terms of RMSE, ρ_{prs} and ρ_{spr}. For the FCE-PUB test set, it performs better than the baseline on RMSE but not the other two metrics. We suggest that the reason is again the difference between the development set and the test set in the FCE-PUB dataset. Concerning the significance, W-FEDA on multiple attributes performs significantly better than the baseline on the PET-PRI, FCE-PRI, CAE-PRI test sets for at least one metric.

In summary, W-FEDA on multiple attributes performs better than W-FEDA on a single attribute in four out of five development and test sets. This would suggest that modelling multiple attributes enables our model to learn more commonalities and differences among different texts relative to single attribute model. However, in our study, it is not always the case that modelling on multiple attributes is better than learning from a single attribute (e.g. in the FCE-PUB test set), and modelling multiple attributes is significantly better than the baseline only in some datasets.

5.4.4 Comparison between W-FEDA and U-FEDA

As we have added weighting hyper-parameters into W-FEDA, we now compare W-FEDA with the original unweighted FEDA (U-FEDA) proposed by Daume III (see Section 5.3.1).

5.4.4.1 Single Attribute

Table 5.9 presents the comparisons between W-FEDA and U-FEDA on all single attributes. In this table, the six sub-tables follow the same organisation as in Table 5.12. For the

Method	ATTR	dev_rmse	dev_prs	dev_spr	test_rmse	test_prs	test_spr
2. PET-PRI							
baseline	-	2.247	0.594	**0.541**	2.179	0.567	0.467
1-best	GENRE	2.214	**0.611**	**0.541**	2.139	0.591	0.491
1-oracle	PROMPT	2.216	0.601	0.540	2.121	0.597	0.494
all		**2.202**	0.609	0.535	**2.094+**	**0.615+**	**0.510**
3. FCE-PRI							
baseline	-	1.994	0.379	0.363	1.991	0.359	0.339
1-best	PROMPT	1.968	0.399	0.379	1.960	0.369	0.350
1-oracle	PROMPT	1.968	0.399	0.379	1.960	0.369	0.350
all	-	**1.954**	**0.415**	**0.385**	**1.952**	**0.387**	**0.368**
4. CAE-PRI							
baseline	-	2.421	0.377	0.341	2.405	0.411	0.410
1-best	PROMPT	2.381	0.412	**0.394+**	2.398	0.436	0.435
1-oracle	TASK	2.417	0.385	0.346	2.368+	0.439	0.438
all	-	**2.379+**	**0.419+**	0.388+	**2.358+**	**0.448**	**0.446**
5. IELTS							
baseline	-	0.701	0.697	0.680	0.693	0.684	0.659
1-best	TASK	**0.694**	**0.705**	**0.692**	0.689	0.693	0.670
1-oracle	GENRE	0.700	0.698	0.684	0.686	0.696	0.670
all	-	0.699	0.702	0.684	**0.684**	**0.701**	**0.675**
6. FCE-PUB							
baseline	-	2.402	0.567	0.571	2.569	0.662	0.652
1-best	TASK	2.373	0.582	0.579	2.556	**0.687**	**0.677**
1-oracle	GENRE	2.378	0.572	0.576	**2.548**	0.661	0.659
all	-	**2.353**	**0.586**	**0.584**	2.564	0.654	0.636

Table 5.8: W-FEDA on multiple attributes, compared to W-FEDA on single attributes and training a single baseline grader. The best results in each dataset are highlighted. + means the corresponding model is significantly better than the baseline by the permutation test with 2,000 samples ($p < 0.05$)

first column *Method*:

- *Baseline*: the support vector regression model described in Chapter 3;
- *U-FEDA*: the U-FEDA trained for each dataset on each single attribute, with all weighting hyper-parameters fixed to 1.0;
- *W-FEDA*: the W-FEDA described in Section 5.3.4.

Column *Beat BASE?* refers to whether the corresponding model performs better than the baseline in terms of RMSE, while column *W-FEDA wins?* presents whether the W-FEDA is better or worse than the U-FEDA in terms of RMSE.

In Table 5.9, it is clear to see that not in every case that W-FEDA is better than U-FEDA. For instance, in terms of RMSE, W-FEDA is better than U-FEDA on:

- the KET-PRI, FCE-PRI and IELTS datasets with the prompt attribute;
- the PET-PRI and FCE-PUB datasets with the genre attribute;
- the CAE-PRI dataset with the task attribute.

In contrast, also in terms of RMSE, U-FEDA performs better than W-FEDA for the PET-PRI dataset for prompt and the IELTS dataset for genre and task. Also, W-FEDA and U-FEDA show equivalent performance on the CAE-PRI dataset for prompt and on the FCE-PUB dataset for prompt & task, as the optimal weighting hyper-parameter found is 1.0.

In terms of statistical significance, there are more metrics and datasets for which W-FEDA performs significantly better than on than U-FEDA.

The mixed performance could be due to the fact that having a wider space to explore hyper-parameters in order to improve performance on the development set, the model might also sometimes overfit to the development set, leading to poorer performance on the test set.

To numerically understand the differences between U-FEDA and W-FEDA in Table 5.9, we count the number of cases in which W-FEDA shows better, worse or the same performance with U-FEDA and report them in Table 5.10. In Table 5.10, differences are given in terms of each evaluation measure.

On the basis of Table 5.10, we can conclude that W-FEDA shows improved performance relative to U-FEDA more frequently than it shows to degraded performance (19 versus 11), although the gap is not large.

To further illustrate the performance difference between W-FEDA and U-FEDA, Table 5.11 summaries the number of times W-FEDA and U-FEDA are better or worse than the baseline. The layout of Table 5.11 is similar to that of Table 5.10, with the three

Method	ATTR	test_rmse	test_prs	test_spr	Beat BASE?	W-FEDA wins?
1.KET-PRI						
Baseline	NONE	0.968	0.706	0.648	-	-
U-FEDA	PROMPT	0.940	0.730	0.685	Yes	Win
W-FEDA		**0.904***	**0.751***	**0.723***	Yes	
2.PET-PRI						
Baseline	NONE	2.179	0.567	0.467	-	-
U-FEDA	PROMPT	**2.108**	**0.608**	**0.499**	Yes	Lose
W-FEDA		2.121	0.597	0.494	Yes	
U-FEDA	GENRE	2.193	0.558	0.459	No	Win
W-FEDA		**2.139***	**0.591***	**0.491**	Yes	
3.FCE-PRI						
Baseline	NONE	1.991	0.359	0.339	-	-
U-FEDA	PROMPT	2.001	0.347	0.322	No	Win
W-FEDA		**1.960***	**0.368**	**0.350**	Yes	
U-FEDA	TASK	**1.964**	0.373	0.352	Yes	Lose
W-FEDA		1.979	**0.374**	**0.354**	Yes	
4.CAE-PRI						
Baseline	NONE	2.405	0.411	0.410	-	-
U-FEDA	PROMPT	2.398	0.436	0.435	Yes	Tie
W-FEDA		2.398	0.436	0.435	Yes	
U-FEDA	TASK	2.387	0.427	0.436	Yes	Win
W-FEDA		**2.368**	**0.439**	**0.438**	Yes	
5.IELTS						
Baseline	NONE	0.693	0.684	0.659	-	-
U-FEDA	PROMPT	0.725	0.648	0.617	No	Win
W-FEDA		**0.703***	**0.666***	**0.633***	No	
U-FEDA	GENRE	**0.680**	**0.700**	**0.679**	Yes	Lose
W-FEDA		0.686	0.696	0.670	Yes	
U-FEDA	TASK	**0.681**	**0.700**	**0.674**	Yes	Lose
W-FEDA		0.689	0.693	0.670	Yes	
6.FCE-PUB						
Baseline	NONE	2.569	0.662	0.652	-	-
U-FEDA	PROMPT	2.662	0.660	0.655	No	Tie
W-FEDA		2.662	0.660	0.655	No	
U-FEDA	GENRE	2.589	**0.664**	0.649	Yes	Win
W-FEDA		**2.548**	0.661	**0.659**	Yes	
U-FEDA	TASK	2.556	0.687	0.677	Yes	Tie
W-FEDA		2.556	0.687	0.677	Yes	

Table 5.9: Comparison between U-FEDA and W-FEDA on the test sets. Highlighted numbers indicate the better results between the U-FEDA and W-FEDA models. * means that W-FEDA is significantly better than U-FEDA expanded on the same attribute using a permutation test with 2,000 samples. ($p < 0.05$)

	RMSE	ρ_{prs}	ρ_{spr}	Total
Win	**6**	**6**	**7**	**19**
Tie	3	3	3	9
Lose	4	4	3	11

Table 5.10: Distribution of performance differences between W-FEDA and U-FEDA on the test sets for each single attribute. The highlighted numbers represent the dominant numbers in each column.

	RMSE		ρ_{prs}		ρ_{spr}		Total	
	W	U	W	U	W	U	W	U
Win	**11**	8	**10**	9	**12**	9	**33**	26
Lose	2	5	3	4	1	4	6	13

Table 5.11: Distribution of W (W-FEDA) and U (U-FEDA) relative to the baseline on the test sets. The highlighted numbers represent the method (W-FEDA or U-FEDA) that more frequently beats the baseline in each column.

columns RMSE, ρ_{prs} and ρ_{spr} representing the performance of W/U-FEDA, relative to the baseline, and the column *Total* providing the sum of the three columns.

On the basis of the data presented in Table 5.11, we conclude that W-FEDA performs better than the baseline more frequently than U-FEDA does in terms of RMSE, ρ_{prs} and ρ_{spr}. Finally, the Total column shows more cases in which W-FEDA outperforms baseline, relative to U-FEDA (33 versus 26).

5.4.4.2 Multiple Attributes

We also measure the performance differences between W-FEDA and U-FEDA on multiple attributes in Table 5.12. The organisation of Table 5.12 follows the presentation of Table 5.9. Similar to Table 5.9, Table 5.12 shows that W-FEDA is not always better than U-FEDA.

The *W-FEDA wins?* column in Table 5.9 is also summarised in Table 5.13. Similar to Table 5.9, Table 5.13 shows that there are more cases in which W-FEDA is better than U-FEDA in terms of RMSE and ρ_{prs} (3 versus 1), and that W-FEDA shows similar performance to U-FEDA on ρ_{spr} (2 versus 2). Finally, the *Total* column shows that there are more cases in which W-FEDA outperforms U-FEDA.

Table 5.14 summarises the number of times of W-FEDA and U-FEDA on multiple attributes beat the baseline. W-FEDA is better than the baseline in four cases; U-FEDA outperforms the baseline three times. This trend is similar to that presented in Table 5.11 on each single attribute that there is one more time that W-FEDA is better than the baseline compared to U-FEDA.

Method	test_rmse	test_prs	test_spr	Better than BASE?	W-FEDA wins?
2.PET-PRI					
Baseline	2.179	0.567	0.467	-	-
U-FEDA	**2.081**	**0.623**	**0.514**	Yes	Lose
W-FEDA	2.094	0.615	0.510	Yes	
3.FCE-PRI					
Baseline	1.991	0.359	0.339	-	-
U-FEDA	1.952	0.387	0.368	Yes	Tie
W-FEDA	1.952	0.387	0.368	Yes	
4.CAE-PRI					
Baseline	2.405	0.411	0.410	-	-
U-FEDA	2.379	0.430	0.426	Yes	Win
W-FEDA	**2.358**	**0.448**	**0.447**	Yes	
5.IELTS					
Baseline	0.693	0.684	0.659	-	-
U-FEDA	0.700	0.677	0.646	No	Win
W-FEDA	**0.684**	**0.701***	**0.676***	Yes	
6.FCE-PUB					
Baseline	2.569	0.662	0.652	-	-
U-FEDA	2.765	0.648	**0.651**	No	Win
W-FEDA	**2.564***	**0.654**	0.636	Yes	

Table 5.12: W-FEDA and U-FEDA on multiple attributes. Highlighted numbers indicate whether U-FEDA or W-FEDA has a better performance. * means that W-FEDA is significantly better than U-FEDA by a permutation test with 2,000 samples at the corresponding metric ($p < 0.05$)

	RMSE	ρ_{prs}	ρ_{spr}	Total
Win	**3**	**3**	**2**	**8**
Tie	1	1	1	3
Lose	1	1	2	4

Table 5.13: Distribution of the performance difference between W-FEDA and U-FEDA on the test sets on all available attributes. The highlighted number in each evaluation metric represents the dominant case in that metric.

	RMSE		ρ_{prs}		ρ_{spr}		Total	
	W	U	W	U	W	U	W	U
Win	**5**	3	**4**	3	**4**	3	**13**	9
Lose	0	2	1	2	1	2	2	6

Table 5.14: Distribution of the performance differences of W (W-FEDA) and U (U-FEDA) relative to the baseline on the test sets. The highlighted number in each evaluation metric represents that method (W-FEDA or U-FEDA) that beats the baseline more frequently.

In summary, although the trend in the performance difference between W-FEDA and U-FEDA is mixed, we found that there are more cases in which W-FEDA outperforms U-FEDA. In addition, hyper-parameter tuning provided performance improvement (even if only marginal) or did not hurt the system performance in more than half of the cases. Nevertheless, the improvements are small in most cases, and they did not happen across all the datasets we evaluated.

5.5 Combined Model based on Authorship Knowledge and Intra-Exam Properties

In this section, we concatenate the vectors based on authorship knowledge used in Chapter 4 and the vectors of W-FEDA in this chapter together to investigate whether combining these two types of meta-properties and training one combined model can further improve model performance or not.

To reduce the hyper-parameters searching space, for each dataset, we choose the best vectors $\Phi_{author}(x)$ and $\Phi_{mdl}(x)$ tuned on the development set from each chapter, and we concatenate these two vectors together via

$$\Phi_{am}(x) = \lambda_a \Phi_{author}(x) \oplus (1 - \lambda_a)\Phi_{mdl}(x) \tag{5.13}$$

where $\lambda_a \in [0.0, 1.0]$ is the hyper-parameter to decide the contribution of $\Phi_{author}(x)$ in the concatenated vector. We train an SVR model on the concatenated vector $\Phi_{am}(x)$ and tune its hyper-parameters. We choose the hyper-parameters with the best performance on each development set and report model performance on the corresponding test set. As only FCE-PRI, CAE-PRI, IELTS and FCE-PUB have more than one response written by each learner, we only report the performance of these four datasets in Table 5.15, and the authorship-based vectors $\Phi_{author}(x)$ are chosen from the feature fusion methods (FF-NT and FF-CT). We will combine the results based on different meta-properties in a score fusion way in Chapter 7.

In terms of the performance in Table 5.15, the combined method only achieves

setup	test_rmse	test_prs	test_spr	λ_a
3. FCE-PRI				
BASE	1.991	0.359	0.339	X
MDL	**1.952**	**0.387**	**0.368**	X
FF-NT	1.982	0.348	0.324	X
FF-CT	1.974	0.354	0.333	X
FF-NT + MDL	1.969	0.380	0.356	0.80
FF-CT + MDL	1.979	0.368	0.342	0.80
4. CAE-PRI				
BASE	2.405	0.411	0.410	X
MDL	2.358+	0.448+	0.446+	X
FF-NT	2.356+	0.460+	0.455+	X
FF-CT	2.360	0.447	0.440	X
FF-NT + MDL	**2.316+**	**0.486+**	**0.472+**	0.60
FF-CT + MDL	2.341+	0.468+	0.467+	0.80
5. IELTS				
BASE	0.693	0.684	0.659	X
MDL	0.684	0.701+	0.675	X
FF-NT	0.683	0.698	0.680	X
FF-CT	**0.664+**	**0.720+**	**0.710+**	X
FF-NT + MDL	0.668+	0.713+	0.697+	0.80
FF-CT + MDL	**0.664+**	**0.720+**	**0.710+**	1.00
6. FCE-PUB				
BASE	2.569	0.662	0.652	X
MDL	2.564	0.654	0.636	X
FF-NT	2.529	0.688	0.688	X
FF-CT	**2.460+**	**0.694**	**0.695**	X
FF-NT + MDL	2.524	0.684	0.688	0.90
FF-CT + MDL	**2.460+**	**0.694**	**0.695**	1.00

Table 5.15: Results of combining the feature fusion and multi-domain learning methods based on multiple attributes together on the test sets. The numbers highlighted are the best in each dataset.

the best performance on the CAE-PRI dataset (FF-NT + MDL). The reason might be that this is only dataset where both multi-domain learning (MDL) and feature fusion (FF-NT) methods are significantly better than the baseline, and both methods find useful information not included in only modelling authorship knowledge or intra-exam meta properties.

MDL achieves the best performance on the FCE-PRI dataset. The reason that MDL is better than the combined method in this case might be the authorship-knowledge based methods perform worse than the baseline in terms of the Pearson and Spearman correlations, and the combined method absorbs the noise from the authorship-knowledge based methods.

The feature fusion with concatenated text (FF-CT) and the combined methods (FF-CT + MDL) achieve the same performance on the IELTS and FCE-PUB datasets, because the hyper-parameter tuned on the development set is 1.0.

In terms of the concatenating hyper-parameter λ_a for each combined method, λ_a is always bigger than 0.5, which shows that the combined method prefers the information from the authorship-knowledge base methods than MDL.

5.6 Discussion

As Section 5.4 shows, although multi-domain learning on different intra-exam properties via W-FEDA can improve model performance, this improvement is not present in every dataset, and only a few models achieve a statistically significant improvement according to the permutation test. Furthermore, in Section 5.5, the combined model favours authorship knowledge over the intra-exam properties based on the value of λ_a. In this section, we present some possible reasons for the results we observed.

First, we can figure out one reason why FEDA does not work on some attributes taken from previous work. In the original FEDA work, Daume III (2007) evaluated his method on tasks including named entity recognition, POS tagging, recapitalisation and treebank chunking. Daume III showed that FEDA improved model performance on named entity recognition, part-of-speech tagging and recapitalisation task, but not treebank chunking. More specifically, FEDA always underperformed the baseline on the Brown corpus for each individual section, and the baseline only mixed all the data instances from all domains together without explicitly modelling any domain knowledge. One reason for the FEDA performance degradation might be related to the fact that the features are not sufficiently discriminative across different domains that the features for tree chunking the data from different domains might be very similar, and the benefit of augmenting feature space for similar domains is not huge. This might also be the case in ATS. For example, some learners' numerous misspelling and grammatical

> Next month, Pete, an English friend who lives in London, will be on holiday in New York for two weeks.
> You have agreed to look after his flat for him while he is away. Read the extract from his letter and your notes.
> Then, using all your notes, write a letter to Pete. ...

Figure 5.10: First prompt from 0102_2001_3.xml of the FCE-PUB dataset which asks learners to write a letter to your friend Pete. The full prompt is available in Appendix B

> You recently had a week's holiday in London.
> During your stay you went to the theatre to see a musical show and had a very disappointing evening.
> Read the advertisement for the show and the notes you have made. Then, write a letter to the manager of the theatre, explaining what the problems were and asking for some money back.
> ...
> Write a letter of between 120 and 180 words in an appropriate style on the opposite page.
> Do not write any postal addresses.

Figure 5.11: First prompt from 0100_2001_6.xml of the FCE-PUB dataset, which requires learners to express their disappointment to a theatre manager. The full prompt is available in Appendix B

errors in their texts may have hampered examiners' understanding of those texts, and the negative contribution of these errors might not be very different across different prompts, genres and tasks in scoring. Therefore, modelling domain information for these kinds of features might not be helpful enough in some cases.

Another possible reason is we have many different attribute values to explicitly model, but the number of texts for some attributes might be relatively low. This might make the model do not have enough texts to learn the correct and useful patterns for some domain-specific features, and overfitting occurs in some datasets.

Furthermore, some attributes not explicitly modelled in any dataset might still be crucial in ATS. These attributes might include topic, formality, and sentiment.

With respect to the topic attribute, prompts relating to the same topic (e.g. education or politics) might share more features relative to other prompts. Learners who use examples from the same topic to support their views might receive higher scores than those who use examples from prompts on a different topic.

As for the formality attribute, it might be important to force learners to use the appropriate register under different circumstances. Two example prompts are provided in Figures 5.10 and 5.11. The prompt in Figure 5.10 asks learners to write a letter to a friend, and the writing style might be relatively informal. In contrast, the prompt in Figure 5.11 requires learners to write a complaint to a music theatre manager. This letter

You see this announcement in an English language magazine.
An unforgettable birthday
Have you ever had an unforgettable birthday?
How did you celebrate it?
What made it so memorable?
Write us an article answering these questions.
The best article will be published in the magazine.
Write your article.

Figure 5.12: Third prompt from 0102_2001_6.xml of the FCE-PUB dataset, which requires learners to write an article describing a birthday memory.

should be written in a formal way and should, for example, contain fewer phrasal verbs, which are considered informal (Williams, 2008, p. 22).

As regards the sentiment attribute, human examiners might expect learners to use appropriate words to reflect the correct sentiment polarity in their text. Two examples are given in Figures 5.11 and 5.12. In Figure 5.11, the prompt asks learners to express their disappointment about a show, so there should be some phrases to correctly express the learners' negative feelings. In comparison, the prompt in Figure 5.12 asks learners to write an article describing a memorable experience about their birthday, so the sentiment here might be relatively positive. It might be useful to divide prompts according to their different sentiment polarities and encourage the prompts in each group to share more parameters.

Finally, other features might be representative but not included in our model. For instance, (Williams, 2008, p. 45) has summarised the key criteria to get a high score on the FCE exam (i.e. the FCE-PUB dataset). One criterion is that learners should use appropriate techniques that are relevant to the task. For instance, in prompts requiring learners to write an article, learners should appropriately use direct quotations from the prompt. Similarly, when composing a letter, learners should skilfully use the letter writing conventions, including salutations, dividing a letter into paragraphs and closing phrases. Although the lexical features described in Section 3.4.2 (e.g. word unigrams and bigrams) might implicitly capture direct quotations and the correct use of salutations, we do not use explicit features that are directly related to these requirements to force the model to capture them. This problem might apply to other question types within the FCE-PUB and other datasets.

5.7 Summary

In this chapter, we investigated various attributes to classify texts into different domains. These attributes include prompt, genre and task. We found that using a technique of frustratingly easy domain adaptation improves the ATS model in more than half of the cases. In addition, multi-domain learning on multiple attributes improves on each single attribute and on not doing multi-domain learning in four out of five test sets. Our comparison of W-FEDA and U-FEDA shows that there are more cases that W-FEDA is better than U-FEDA at beating the baseline, although the difference is not large. Finally, we provided some possible explanations as to why W-FEDA only significantly improves model performance in the observed cases.

Chapter 6

Automated Scoring across Different Exams by Transfer Learning

6.1 Introduction

Every English writing exam has its own grading criteria; however, these exams can also share commonalities in their grading criteria. One interesting question to investigate is whether we can combine datasets from different sources to train an ATS model. The benefit of doing this might be having more data to train an ATS system more robustly, potentially giving a performance improvement. Such a system might be better than a single system only trained on the data from one source (Dong et al., 2015; Luong et al., 2016).

As we have discussed in Section 2.5.3, although Phandi et al. (2015) and Cummins et al. (2016b) have investigated how to combine the essays from different exams to train a single model, they both only used the ASAP dataset. One problem of the ASAP dataset is that this dataset only contains essays written by English native speakers. Besides this, this dataset was collected from learners at similar proficiency levels so that the grade range of the participating students only lies between Grade 7 to Grade 10 in the United States. Furthermore, for each specific work, Phandi et al. only studied domain adaptation between two groups of essays on similar topics; it is not clear whether domain adaptation from more than two or more target proficiency levels works well. In comparison, Cummins et al. built a single ranking-based model for all the exams in the ASAP dataset. Whether the same multi-task learning setup works for all the exams from distinct proficiency levels was not examined.

To make the terminology in this chapter clear, in transfer learning, when we build a model to mark the texts from a single dataset, we only use the texts from other datasets to optimise model performance on this single dataset. This single dataset is the *target dataset*; the other datasets excluding the target dataset are the *source datasets*.

The goal of this chapter is to study whether existing transfer-learning techniques can boost grader performance on any target non-native English learners' dataset (Chapter 3) from different target proficiency levels. More specifically, whether bringing non-native learners' data from multiple source datasets can affect grader performance on the target dataset. We also intend to implement and improve existing transfer-learning techniques to improve grader performance on the target dataset.[1]

6.2 Models

In this section, we describe the models used in this chapter which leverage multiple datasets from various sources.

6.2.1 Two-Stage Model

We can use the regression-based weighted frustratingly easy domain adaptation (W-FEDA) model from Chapter 5 to utilise multiple datasets. However, there is a potential drawback of this model, which is the incompatibility of different exam marking schemes. For example, the meaning of a score of 16 in the PET exam is different from a score of 16 in the FCE exam. When we are training a model to predict the scores of the texts for PET, the model might be confused by the same numerical scores from different marking schemes. Although W-FEDA might overcome this problem by having a separate domain-specific weight parameter space for each dataset, the shared parameter space shared by different datasets might still be a compromise between the absolute scores from different marking schemes.

Hence, we suggest that when training an transfer-learning based automated grader on multiple datasets from different sources to mark the target dataset, it is better to transfer only the ranking of the language quality among the texts from different source datasets, rather than the absolute scores. The benefit of transferring ranking is that when we put different datasets together, we do not need to worry about how to handle the existence of different marking schemes.

We make a strong assumption here that given any pair of texts $text_1$ and $text_2$, if the score of $text_1$ is higher than $text_2$ on one grading scale, the relative order between these two texts is still preserved even if we use a new marking scheme to mark these two texts. Based on this assumption, transferring the relative order might bring less noise and more useful knowledge to be passed into the shared parameter space from different

[1] The idea of the two-stage model in Section 6.2 was published as a paper in the proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Cummins et al., 2016b). In this paper, I contributed the idea of the two-stage model. Cummins implemented the model and evaluated it on the ASAP dataset. Briscoe gave feedback and suggestion for the paper.

datasets as the order is always correct in all scales. Cummins et al. (2016b)'s model also utilises multiple datasets by transferring relative order instead of absolute scores. In the following section, we will describe their work in detail.

6.2.1.1 Cummins et al.'s Model

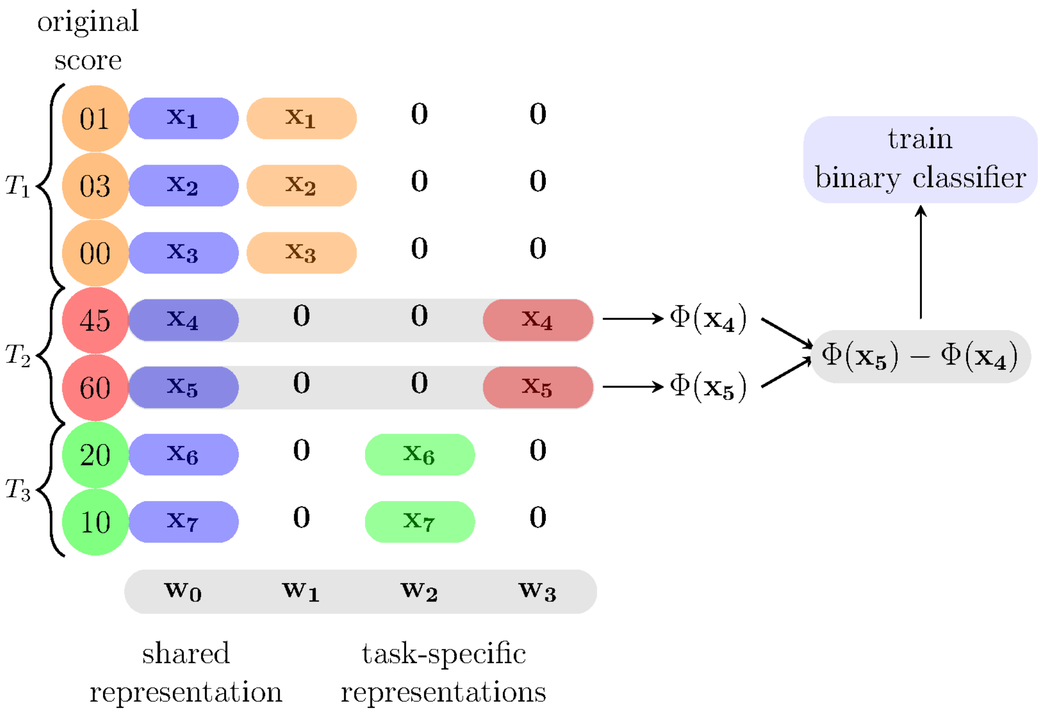


Figure 6.1: First stage of Cummins et al.'s model.

In Cummins et al. (2016b)'s original work in Figure 6.1, we assume that we have seven texts with their feature vectors from x_1 to x_7. These texts come from three different tasks T_1, T_2, and T_3 with three different grading scales (the orange, red and green numbers in Figure 6.1). Each text is marked with the grading scale of the task which the text belongs to[2], and the feature vector for each text is x_i. We use unweighted FEDA (Section 5.3.1) to augment each feature vector to $\Phi(x_i)$.

The automated scoring procedure of their model is split into two stages. In the first stage, we use a perceptron based pairwise-ranking model (Joachims, 2002), which has been used by Briscoe et al. (2010); Yannakoudakis (2013) (Section 2.4) in ATS to do ranking. The model constraints during training are similar to the ranking SVM (Section 2.3.3) in the following equation:

$$\mathbf{w}^T(\Phi(x_i) - \Phi(x_j)) > M \; for \; (x_i, x_j) \in r \tag{6.1}$$

[2] The definition of a task given by Cummins et al. (2016b) is different from the task we define in Chapter 5. The task here is similar to a prompt in this thesis. Different tasks could be marked on the same grading scale or different scales.

In the above equation, $(x_i, x_j) \in r$ represents all the data instance pairs in the training set where x_i has an higher rank compared to x_j. During training, the model learns a binary classifier (top right in Figure 6.1) with a weight vector $\mathbf{w}$ to minimise the misclassification rate of *difference vectors* $\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{x}_j)$ to ensure most data instance pairs (x_i, x_j) are bigger than M, which is the hyper-parameter to control the model margin. The learned optimal parameters for $\mathbf{w}$ after training is $\mathbf{w}^*$ based on the perceptron algorithm.

Similar to the ranking SVM, during prediction, the perceptron model predicts a *ranking score* $\hat{y}_i^{rank}$ for an incoming data instance x_i by $\hat{y}_i^{rank}$ by $\mathbf{w}^* \cdot \Phi(x_i)$, and we can get the order of an dataset $\{x_i\}_1^N$ via calculating the ranking score for each instance and sort the dataset based on the predicted ranking scores.

After we finish training our ranking model, we still do not know the score of each text on its original grading scale but only an order of all the texts. Therefore, we need another stage to get the predicted score for each text on its corresponding grading scale from the ranking model. In this second stage, for feature vector $\Phi(\boldsymbol{x}_i)$ with predicted ranking score $\hat{y}_i^{rank}$, a linear regression model $\mathbb{R}^1 \rightarrow \mathbb{R}^1$ is built for all the texts from the same task to learn the relation between $\hat{y}_i^{rank}$ and y_i. We use this linear regression model to predict the score of x_i and round it as $\hat{y}_i$. Therefore, we have one linear regression model for each task, and three regression models in total in the example of Figure 6.1.

In summary, we can interpret Cummins et al.'s model as a model combining multi-task learning and transfer learning together. The first stage is a multi-task learning model optimised on all datasets equally, and the second stage distils and transfers the knowledge from the first stage to each task, respectively.

In this chapter, we define a model that learns a ranking model first and then builds a regression model based on the outputs of the ranking model as a *two-stage model*. In contrast, the W-FEDA model described in Section 6.3.1 predict the score of each text in one stage, and we define this type of model as a *one-stage model*.[3]

We slightly modify Cummins et al.'s model in that the first-stage model now is the ranking SVM model in LIBLINEAR (Lee and Lin, 2014) instead of the perceptron model, because the ranking SVM model and their perceptron model have similar performance on ATS (Yannakoudakis, 2013), and the LIBLINEAR ranking SVM implementation is highly optimised for convergence speed, allowing this model to be trained without the need to sample data in order to reduce the training set size. We use the L2-regularisation version of ranking SVM in this chapter.

6.2.1.2 Two-Stage Feature-Rich Model

There is one possible drawback in Cummins et al.'s model. In the first stage, we condense all the features of each text into a ranking score, and these features are no

[3] The models used in Chapter 4 and Chapter 5 are also one-stage models

longer visible in the second stage. In other words, the ranking model is optimised on the order (binary classification error rate) rather than the absolute scores, and the regression model might not have enough information to map the order to the original scores by only seeing a single ranking score. Also, Cummins et al. built one single model for every dataset, and the amount of knowledge that should be transferred from the source datasets to the target dataset to give the best model performance and mitigate the potential negative-transfer influence (Rosenstein et al., 2005) might also vary from one dataset to another.

We propose a variation of Cummins et al.'s model addressing these problems. In the second stage, we hypothesize that having a transfer-learning based model with richer features as the inputs to the model should lead to better performance than Cummins et al.'s original approach. We add the ranking scores predicted by the ranking model as an extra feature $\hat{y}_i^{rank}$ to the features $\mathbf{x}_i$ we identified in Chapter 3 and concatenate them together. In other words, we use the ranking score as an additional feature and feed it into the baseline SVR model in Chapter 3. Here, we include a weighting hyper-parameter $\zeta \in [0, 1]$ to control the influence of the ranking score in the second stage. The feature representation of the second stage to train the regression model can be written as:

$$\mathbf{x}_i^{FR} = \zeta \mathbf{x}_i \oplus (1 - \zeta)\hat{y}_i^{rank} \tag{6.2}$$

All the predicted ranking scores $\hat{y}^{rank}$ are normalised to $[0, 1]$ to ensure they are on the same scale with the features to be concatenated, and we use the SVR model in Chapter 3 as the second-stage model to learn the relation between the new feature vectors $\mathbf{x}_i^{FR}$ and the score y_i.

6.3 Transfer-Learning Techniques

In this section, we introduce the transfer-learning techniques we will implement and describe how to put them into the one-stage (defined in Section 6.2.1.1) and two-stage models in Section 6.2.1.

6.3.1 W-FEDA

Based on W-FEDA described in Section 5.3.4, we tune the weighting hyper-parameter for all the texts from all the datasets so that we can weigh how important each space is in W-FEDA. More specifically, we rewrite Equation 5.4 into the following form to accommodate the transfer-learning requirement by adding extra hyper-parameters $\boldsymbol{\eta}$:

$$\Phi^{\text{aug}}(x) = \oplus_{a=0}^{k} \eta_a f(x, a) \tag{6.3}$$

and $f(x, j)$ is still defined as it was in Equation 5.4:

$$f(x, a) = \begin{cases} \mathbf{x}, & \text{if } a = 0 \\ \mathbf{x}, & \text{if Domain}(x) = a \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{6.4}$$

η_a is set to 1.0 if attribute $a = td$ corresponds to the target dataset *td*. $\boldsymbol{\eta}/\eta_{td}$ for the shared space and other source datasets are tuned between $(0.0, 1.0)$. This W-FEDA model is evaluated as one-stage transfer-learning model. We set these hyper-parameters to the same value to reduce the search space.

For the two-stage models, we only apply this approach to the first stage of each model. This is similar to what Cummins et al. (2016b) did except that they used standard FEDA and set $\eta_a = 1.0$ for every task a in the first stage.

Although the performance difference between W-FEDA and unweighted FEDA in Chapter 5 is mixed, we still use W-FEDA in this chapter, because W-FEDA is more suitable in modelling the differences across multiple datasets as it has the weighting hyper-parameters to adjust the attention to each dataset, and the differences between various datasets are much bigger than the difference within the same dataset. Therefore, W-FEDA can help us put more attention and priority on the target dataset.

6.3.2 Instance Weighting

Another approach to utilise the data from different sources is instance weighting. Jiang and Zhai (2007) (Section 2.5.2) proposed an approach to combine different datasets via instance weighting. Following their strategy, we split the training dataset $\{(x_i, y_i)\}_{i=1}^{N}$ into the target training set $\{(x_{j,td}, y_{j,td})\}_{j=1}^{N_{td}}$ and the source training set $\{(x_{i,sd}, y_{i,sd})\}_{i=1}^{N_{sd}}$, and N_{sd} and N_{td} represent the number of data instances from the source and target datasets, respectively. The cost hyper-parameter C of the support vector regression (SVR) model in Equation 2.24 is also split into C_{td} for the target dataset *td* and C_{sd} for the other source datasets *sd* by:

$$
\begin{aligned}
\min_{\mathbf{w},b,\xi_{td}^{upper},\xi_{td}^{lower},\xi_{sd}^{upper},\xi_{sd}^{lower}} \quad & C_{sd}\textstyle\sum_{i=1}^{N_{sd}}((\xi_{i,sd}^{upper})^2 + (\xi_{i,sd}^{lower})^2) + C_{td}\sum_{j=1}^{N_{td}}((\xi_{j,td}^{upper})^2 + (\xi_{j,td}^{lower})^2) + \frac{1}{2}\|\mathbf{w}\|^2 \\
\text{subject to :} \quad & \begin{cases}
\left.\begin{aligned} & y_{i,sd} - \mathbf{w}^T\mathbf{x}_{i,sd} - b \leq \epsilon + \xi_{i,sd}^{upper} \\ & \mathbf{w}^T\mathbf{x}_{i,sd} + b - y_{i,sd} \leq \epsilon + \xi_{i,sd}^{lower} \\ & \xi_{i,sd}^{upper}, \xi_{i,sd}^{lower} \geq 0 \end{aligned}\right\} & \forall i \in 1,\ldots,N_{sd} \\
\left.\begin{aligned} & y_{j,td} - \mathbf{w}^T\mathbf{x}_{j,td} - b \leq \epsilon + \xi_{j,td}^{upper} \\ & \mathbf{w}^T\mathbf{x}_{j,td} + b - y_{j,td} \leq \epsilon + \xi_{j,td}^{lower} \\ & \xi_{j,td}^{upper}, \xi_{j,td}^{lower} \geq 0 \end{aligned}\right\} & \forall j \in 1,\ldots,N_{td}
\end{cases}
\end{aligned}
\tag{6.5}
$$

We set C_{td} larger than C_{sd} so that the model pays more attention to the data instances from the target dataset. In this chapter, we combine this weighting scheme with standard FEDA and add it to the one-stage models and the first stage of all the two-stage models to be implemented.

6.4 Implementation

In this section, we describe the implementation of the models we have illustrated in Section 6.2.

6.4.1 Baseline Models Trained on One Dataset

The individual-level baseline model (BASE) is still an SVR model based on LIBLINEAR as used in Chapter 3. The cost hyper-parameter C is tuned on each development set, and the epsilon hyper-parameter is set to the default value of 0.1 in LIBLINEAR.

We also implement a ranking-based model (BRank) as the second baseline model. We train an SVM pairwise-ranking model only on the training set of each dataset, tune the model on its development set, and evaluate the model on its test set. The reason we list this pairwise-ranking model as another baseline model is that we want to verify whether the performance difference between the two-stage models trained on multiple datasets and the baseline model is actually affected by bringing more data from various sources if there is any difference. In other words, if there is any performance improvement or degradation, it can be a consequence of the ensemble of different models rather than bringing more datasets from various sources (Mitchell, 1997, pp. 175) (Joshi et al., 2012).

Furthermore, we have a two-stage model as the third baseline in this chapter. It uses BRank as the first stage model and BASE as the second stage model. We can use this model to further measure how much of the performance variation of the two-stage model utilising multiple datasets is due to the effect of ensemble learning.

6.4.2 Transfer-Learning Models Trained on Multiple Datasets

For a transfer-learning model, we train each model by selecting one of the six training sets as the target dataset and use the rest as the source datasets. Each model is then tuned and tested on the target development and test sets respectively. Hence, for each transfer-learning setup, we have one model for each dataset, and six models in total.

For a one-stage transfer-learning model, to ensure the scores for each dataset are approximately in the same range, we multiply the score of each text from the KET-PRI dataset by 4, and the scores of the IELTS dataset by 2. We then convert the scores of these datasets back to their corresponding original grading scales when we report the results.

Cummins et al. did not conduct any hyper-parameter tuning and only used the default hyper-parameters. As we have a development set for each dataset, we can do hyper-parameter tuning. Here we report how to conduct the hyper-parameter tuning:

- For the one-stage models, we tune all the hyper-parameters on the development sets.
- For the two-stage models for each target dataset:
    - In the first stage, we tune the ranking SVM cost hyper-parameters for the target and source datasets. We also tune the weighting hyper-parameters for W-FEDA and instance weighting in the first stage. We then pick the model with the hyper-parameters achieving the highest Pearson correlation on the target development dataset.
    - In the second stage, we tune the SVR cost hyper-parameter on the target development set. For the two-stage feature rich models, we also tune the interpolation hyper-parameter described in Section 6.2.1.2.

6.5 Results

In this section, we report the results of the three baseline models and all the transfer-learning models we have built. We evaluate these models on all the development sets. For the test sets, we report the three baseline models as well, and we select the transfer-learning model achieving the lowest root-mean-squared error (RMSE) on each development set and evaluate it on the corresponding test set.

The results of all the models evaluated on the development sets are reported in Table 6.1 and Table 6.2 with RMSE, and Pearson (ρ_{prs}) and Spearman (ρ_{spr}) correlations, and the p-values of the Wilcoxon signed-rank test across all the six development sets are reported in Table 6.3.

Let us explain the abbreviations of the model names in Table 6.1, Table 6.2 and Table 6.3 before we start to discuss the results:

- Baseline models:
    - *BASE*: it is the baseline SVR described in Chapter 3;
    - *BRank*: it is the baseline ranking SVM model described in Section 6.4.1;
- Model types:
    - *OS* (e.g. **OS**-WFEDA): it is a one-stage regression model;
    - *TS* (e.g. **TS**-IW-FR): it is a two-stage model;
- Transfer-learning strategies:
    - *WFEDA* (e.g. OS-**WFEDA**): the corresponding model uses W-FEDA as the transfer-learning strategy;
    - *IW* (e.g. OS-**IW**): the corresponding model uses instance weighting with standard FEDA as the transfer-learning strategy;
- Second stage of the two-stage models:
    - *ONE* (e.g. TS-IW-**ONE**): it is a two-stage model that uses only one rank score to predict the original score of a text at the second stage. This is the same strategy used by Cummins et al.;
    - *FR* (e.g. TS-IW-**FR**): it is a feature-rich two-stage model. This is the variant we propose in this chapter.

6.5.1 Baselines

We first compare the performance differences between the baseline SVR (BASE) and ranking models on the development sets. In terms of RMSE, BRank is worse than BASE on the development sets for KET-PRI, PET-PRI, FCE-PRI and CAE-PRI. Meanwhile, a mixed result is also found on ρ_{prs}. We combine these two models as the two-stage model with baseline ranking model (TS-BRank-FR). For the Wilcoxon test, we can see that there is no significant difference between TS-BRank-FR and BASE.

setup	dev_rmse	dev_prs	dev_spr
1. KET-PRI			
BASE	0.970	0.661	0.585
BRank	0.980	0.655	0.615
TS-BRank-FR	0.970	0.661	**0.619**
OS-WFEDA	0.966	**0.665**	0.593
OS-IW	0.988	0.643	0.569
TS-WFEDA-ONE	0.998	0.640	0.557
TS-IW-ONE	1.023	0.620	0.564
TS-WFEDA-FR	0.970	0.661	0.585
TS-IW-FR	**0.964**	0.664	0.595
2. PET-PRI			
BASE	2.247	0.594	**0.541**
BRank	2.260	0.580	0.526
TS-BRank-FR	2.227	0.595	**0.541**
OS-WFEDA	2.229	0.599	0.540
OS-IW	2.255	0.577	0.507
TS-WFEDA-ONE	2.233	0.592	0.537
TS-IW-ONE	2.253	0.584	0.525
TS-WFEDA-FR	**2.198**	**0.607**	0.532
TS-IW-FR	2.213	0.600	0.520
3. FCE-PRI			
BASE	1.994	0.379	0.363
BRank	2.049-	0.393	0.378
TS-BRank-FR	2.032	0.406	0.386
OS-WFEDA	1.982	0.392	0.382
OS-IW	1.977	0.387	0.377
TS-WFEDA-ONE	2.027	0.410	0.398
TS-IW-ONE	2.027	0.410	0.398
TS-WFEDA-FR	**1.938+**	**0.431+**	**0.424+**
TS-IW-FR	1.973	0.395	0.381

Table 6.1: Results of different setups on the KET-PRI, PET-PRI and FCE-PRI development sets. The best setup per dataset is in **bold**. The green colour means improvement and the red colour means degradation over BASE. + and - mean significantly better and worse ($p < 0.05$) than BASE using the permutation randomisation test (Yeh, 2000) with 2,000 samples.

setup	dev_rmse	dev_prs	dev_spr
4. CAE-PRI			
BASE	2.421	0.377	0.341
BRank	2.495-	0.372	0.346
TS-BRank-FR	2.478-	0.368	0.340
OS-WFEDA	**2.390**	**0.419+**	0.381+
OS-IW	2.410	0.390	0.351
TS-WFEDA-ONE	2.457	0.406	**0.383**
TS-IW-ONE	2.475	0.401	0.371
TS-WFEDA-FR	2.392	0.404	0.371
TS-IW-FR	2.410	0.408	0.371
5. IELTS			
BASE	0.701	0.697	0.680
BRank	0.682+	0.716+	0.698
TS-BRank-FR	0.682+	0.716+	0.700+
OS-WFEDA	0.688	0.712	0.685
OS-IW	0.689	0.709	0.684
TS-WFEDA-ONE	**0.663+**	**0.735+**	**0.712+**
TS-IW-ONE	0.680	0.720	0.700
TS-WFEDA-FR	0.664+	0.734+	0.705+
TS-IW-FR	0.699	0.701	0.686
6. FCE-PUB			
BASE	2.402	0.567	0.571
BRank	2.383	0.570	0.564
TS-BRank-FR	2.373	0.575	0.570
OS-WFEDA	2.387	0.566	0.559
OS-IW	2.392	0.564	0.556
TS-WFEDA-ONE	2.397	0.577	0.567
TS-IW-ONE	2.403	0.573	0.565
TS-WFEDA-FR	**2.362**	**0.585**	**0.577**
TS-IW-FR	2.394	0.573	0.571

Table 6.2: Results of different setups on the CAE-PRI, IELTS and FCE-PUB development sets. The best setup per dataset is in **bold**. The green colour means improvement and the red colour means degradation over BASE. + and - mean significantly better and worse ($p < 0.05$) than BASE using the permutation randomisation test (Yeh, 2000) with 2,000 samples.

setup	dev_rmse	dev_prs	dev_spr
BRank	0.463	0.833	0.293
TS-BRank-FR	0.834	0.208	0.293
OS-WFEDA	0.028	0.046	0.249
OS-IW	0.345	0.753	0.345
TS-WFEDA-ONE	0.753	0.173	0.344
TS-IW-ONE	0.173	0.600	0.463
TS-WFEDA-FR	0.036	0.036	0.142
TS-IW-FR	0.028	0.027	0.295

Table 6.3: p-value for each approach estimated by the Wilcoxon signed-rank test (Demšar, 2006) across all the six development sets in this chapter. The metrics of each setup which are better than the baseline BASE with $p < 0.05$ are highlighted in green.

6.5.2 Transfer-Learning Models

Now we switch to a discussion of the transfer-learning models.

We first look at the one-stage models. The one-stage regression model with weighted FEDA (OS-WFEDA) achieves the best result in terms of RMSE and ρ_{prs} on the CAE-PRI development set, and it is only worse than BASE on the PET-PRI and FCE-PUB development sets in terms of ρ_{spr}. Based on the Wilcoxon test in Table 6.3, OS-WFEDA is significantly better than BASE in terms of RMSE and ρ_{prs}, and no statistically significant difference is detected on the one-stage regression model with instance weighting and standard FEDA (OS-IW).

Finally, we come to the two-stage feature-rich models. In terms of the Wilcoxon test, both TS-WFEDA-FR and TS-IW-FR are significantly better than BASE in terms of RMSE and ρ_{prs}. TS-WFEDA-FR performs better than BASE except for the KET-PRI development set and the PET-PRI development set in terms of ρ_{spr}, and TS-IW-FR is always better than BASE except for the PET-PRI and FCE-PUB development sets in terms of ρ_{spr}.

To better understand how each model performs across all the development sets, we summarise the distribution of how many times each model gets the best performance on each metric in Table 6.4. We can see that two-stage feature-rich WFEDA (TS-WFEDA-FR) deliver the best performance in terms of RMSE, ρ_{prs} and total more frequently than the other approaches, and TS-WFEDA-ONE, TS-WFEDA-FR and TS-BRank-FR have the best performance twice in terms of ρ_{spr}. We also reduce the same models with different transfer-learning strategies (W-FEDA and IW) to the same setup in Table 6.5, and the general pattern is similar to Table 6.4.

Although there are more cases that transferring relative order is better than transferring absolute scores in Table 6.4 and 6.5, we still observe that in some datasets, transferring absolute scores could be more beneficial in terms of some metrics. For

setup	dev_rmse	dev_prs	dev_spr	total
BASE	0	0	1	1
BRank	0	0	0	0
TS-BRank-FR	0	0	**2**	2
OS-WFEDA	1	2	0	3
OS-IW	0	0	0	0
TS-WFEDA-ONE	1	1	**2**	4
TS-IW-ONE	0	0	0	0
TS-WFEDA-FR	**3**	**3**	**2**	**8**
TS-IW-FR	1	0	0	1

Table 6.4: Distribution of how many times each approach achieving the best performance on each metric across all the six development sets. The numbers highlighted are the best in each column. The *total* column is the sum of counts for all three metrics.

setup	dev_rmse	dev_prs	dev_spr	total
BASE	0	0	1	1
BRank	0	0	0	0
TS-BRank-FR	0	0	**2**	2
OS	1	2	0	3
TS-ONE	1	1	**2**	4
TS-FR	**4**	**3**	**2**	**9**

Table 6.5: Reduced distribution based on the transfer learning strategies. Each row represents how many times each approach achieving the best performance on each metric across all the six development sets. The numbers highlighted are the best in each column. The *total* column is the sum of counts for all three metrics.

example, the one-stage model OS-WFEDA has the best performance on the CAE-PRI development set in terms of RMSE and ρ_{prs}. We suggest that the reason might be that for some exams, the standard for achieving the same score might be similar, as for the FCE and CAE exams in Table 6.6.[4,5,6] We can see that the marking criteria to achieve 5 out of 5 in terms of language quality are similar for the FCE and CAE, unlike for the KET exam, because the target proficiency levels for the FCE and CAE exams are similar compared the KET exam. In this case, we suggest that the absolute scores might still have their values in transfer learning as the scores on these two exams are similar.

In Table 6.5, we notice that in four out of six development sets, the two-stage model TS-FR we propose gets the best result in terms of RMSE, and the two-stage model TS-ONE proposed by Cummins et al. also gets the best result in terms of RMSE for one time on the IELTS dataset (TS-WFEDA-ONE in Table 6.2). We suggest that the ranking scores predicted from the first-stage model have different influence on different graders. Some graders on some datasets prefer the knowledge learned from the first-stage model compared to other graders, and they do not need to look back to the original text features at the second stage any more. An example is the IELTS development set, where TS-WFEDA-ONE is better than TS-WFEDA-FR, and the performance difference between TS-WFEDA-ONE and TS-WFEDA-FR is not big. For the other datasets, the original features are still valuable in predicting text scores at the second stage. Without using the original features, we can see that both TS-WFEDA-ONE and TW-WFEDA-IW even get worse on the KET-PRI, FCE-PRI and CAE-PRI datasets in terms of RMSE.

Furthermore, if we reduce all the transfer-learning approaches in Table 6.5 into a single setup and compare this setup with the other three baselines, we can see that the transfer-learning approach dominates all the three metrics in this table. This encourages us to choose the best performing transfer-learning approach on each dataset and evaluate the chosen approach on the test set.

6.5.3 Test sets

Finally, as we have tested multiple setups, for each dataset, we report the setup achieving the lowest RMSE with its corresponding performance on the test sets. We summarise these results in Table 6.7 and the Wilcoxon signed-rank test in Table 6.8.

For the baseline models on the test sets, in terms of RMSE, BRank is worse than BASE on the FCE-PRI, CAE-PRI and IELTS test sets, and TS-BRank-FR is worse than BASE on the FCE-PRI, CAE-PRI and FCE-PUB test sets. In terms of the Wilcoxon test in

[4] http://www.cambridgeenglish.org/images/cambridge-english-assessing-writing-performance-at-level-b2.pdf

[5] http://www.cambridgeenglish.org/images/cambridge-english-assessing-writing-performance-at-level-c1.pdf

[6] http://assets.cambridge.org/97805217/54804/excerpt/9780521754804_excerpt.pdf

Exam	Marking Criteria
CAE (C1)	Uses a range of vocabulary, including less common lexis, effectively and precisely. Uses a wide range of simple and complex grammatical forms with full control, flexibility and sophistication. Errors, if present, are related to less common words and structures, or occur as slips.
FCE (B2)	Uses a range of vocabulary, including less common lexis, appropriately. Uses a range of simple and complex grammatical forms with control and flexibility. Occasional errors may be present but do not impede communication.
KET (A2)	All three parts of message clearly communicated. Only minor spelling errors or occasional grammatical errors.

Table 6.6: Selected marking criteria for the CAE, FCE and KET exams. The first two rows are the marking criteria to get 5 out of 5 in terms of language quality for the CAE and FCE exams. The third row is the marking criterion to get 5 out of 5 regarding the total score for the KET exam. (There is no separate score for language quality for the KET exam.)

Table 6.8, BRank is significantly better than BASE in terms of ρ_{spr}, but not RMSE and ρ_{prs}. In summary, the performance differences between these baselines are mixed.

In contrast, we can see that the transfer-learning approaches we have selected are always better than BASE across all the six test sets, and they are significantly better than BASE on the PET-PRI, CAE-PRI and IELTS datasets on all the three metrics.

We then summarise the performance differences between the transfer-learning techniques and the other two baseline approaches BRank and TS-BRank-FR. The transfer-learning techniques have the best results on the PET-PRI, IELTS and FCE-PUB datasets for all the three metrics, and BRank has the best results on the KET-PRI dataset in terms of ρ_{prs} and ρ_{spr}.

To understand the performance differences more clearly, we summarise Table 6.8 to see the distribution of which approach performs best on each metric. We can see that Transfer-Learning still dominates RMSE, ρ_{prs}, ρ_{spr} and total.

We can conclude that utilising multiple datasets gives us a consistent improvement over the SVR baseline model, and this improvement is not an effect of switching to another type of model or of conducting model ensembles without bringing more training data from different sources.

It is clear that the best approach tuned on the development set is not always significantly better than BASE in all the test sets, especially for the KET-PRI test set where the transfer learning approach is not significantly better than BASE on any metric.

setup	dev_rmse	dev_prs	dev_spr	test_rmse	test_prs	test_spr
1. KET-PRI						
BASE	0.970	0.661	0.585	0.968	0.706	0.648
BRank	0.980	0.655	0.615	**0.959**	0.711	**0.685**
TS-BRank-FR	0.970	0.661	**0.619**	0.964	0.708	0.681
TS-IW-FR	**0.964**	**0.664**	0.595	0.963	**0.713**	0.661
2. PET-PRI						
BASE	2.247	0.594	**0.541**	2.179	0.567	0.467
BRank	2.260	0.580	0.526	2.106	0.604	0.534+
TS-BRank-FR	2.227	0.595	**0.541**	2.112	0.600	0.514+
TS-WFEDA-FR	**2.198**	**0.607**	0.532	**2.045+**	**0.637+**	**0.557+**
3. FCE-PRI						
BASE	1.994	0.379	0.363	1.991	0.359	0.339
BRank	2.049-	0.393	0.378	2.038	0.367	0.347
TS-BRank-FR	2.032	0.406	0.386	2.042	0.365	0.343
TS-WFEDA-FR	**1.938+**	**0.431+**	**0.424+**	**1.951+**	**0.395**	**0.372**
4. CAE-PRI						
BASE	2.421	0.377	0.341	2.405	0.411	0.410
BRank	2.495-	0.372	0.346	2.420	0.424	0.428
TS-BRank-FR	2.478-	0.368	0.340	2.468-	0.373	0.366-
OS-WFEDA	**2.390**	**0.419+**	0.381+	**2.338+**	**0.467+**	**0.468+**
5. IELTS						
BASE	0.701	0.697	0.680	0.693	0.684	0.659
BRank	0.682+	0.716+	0.698	0.697	0.683	0.671
TS-BRank-FR	0.682+	0.716+	0.700+	0.691	0.689	0.676
TS-WFEDA-ONE	**0.663+**	**0.735+**	**0.712+**	**0.669+**	**0.712+**	**0.702+**
6. FCE-PUB						
BASE	2.402	0.567	0.571	2.569	0.662	0.652
BRank	2.383	0.570	0.564	2.560	0.644	0.646
TS-BRank-FR	2.373	0.575	0.570	2.574	0.638	0.642
TS-WFEDA-FR	**2.362**	**0.585**	**0.577**	**2.446**	**0.679**	**0.661**

Table 6.7: Three baselines without using extra data and the transfer-learning models (setups in bold) achieving the lowest RMSE on the development sets. The green colour means improvement and the red colour means degradation over BASE. + and - mean significantly better and worse ($p < 0.05$) than BASE using the permutation randomisation test (Yeh, 2000) with 2,000 samples.

setup	test_rmse	test_prs	test_spr
BRank	0.917	0.345	0.046
TS-BRank-FR	0.752	0.917	0.463
Transfer-Learning	0.028	0.028	0.028

Table 6.8: p-value for each approach estimated by the Wilcoxon signed-rank test (Demšar, 2006) across all the six test sets in this chapter. The metrics of each setup better than the baseline (BASE) with $p < 0.05$ are highlighted in green. The *Transfer-Learning* setup is the combination of different transfer learning approaches achieving the lowest RMSE on the development sets in Table 6.7.

setup	test_rmse	test_prs	test_spr	total
BRank	0	0	0	0
TS-BRank-FR	1	0	1	2
Transfer-Learning	**5**	**6**	**5**	**16**

Table 6.9: Distribution of how many times each approach achieves the best performance on each metric across all the six test sets. The numbers highlighted are the best in each column. The *total* column is the sum of counts for all three metrics.

We suggest there are two reasons.

The first possible reason is that the degree of similarity between datasets varies from one dataset to another, and this degree dictates how much useful knowledge can pass from various source datasets to the grader assessing the target dataset (Rosenstein et al., 2005). For example, the target proficiency levels of some datasets we describe in Chapter 3 are different. The CAE exam is designed to verify whether learners can meet the C1 level, and the KET exam aims at the A2 level. The differences and similarities in the target proficiency levels are also reflected in their grading criteria in Table 6.6. We can clearly see that the criteria for CAE are more similar to FCE compared to those of KET. The writing sections of both CAE and FCE emphasise some higher level aspects including precise vocabulary usage and control of grammatical forms, while KET has no explicit requirement on these two aspects.

The other possible reason is that the ratio between the number of data instances from the source domains to the target domain is not high enough. Cummins et al. (2016b) showed that the performance improvement of utilising multiple datasets to train a grader is higher if there are only limited data instances in the training set from the target dataset but abundant instances from other source datasets. The performance gap might become smaller if we use more training data from the target dataset to train the grader.

To quantitatively study the influence of target data in transfer learning, we remove all the target training set and only use the source datasets as the training set to train a model. As we have no data from target domain to fit the original score scale for each target exam, we only build a ranking model based on W-FEDA and report the Pearson and Spearman's correlations for each new model on the corresponding test set. In this setup during training, we only keep and use the target dataset as the development set for hyper-parameters tuning for the cost hyper-parameter of SVR and the interpolation weights of W-FEDA. Here we set the weight of each source dataset to the same value to reduce the hyper-parameter searching space. Each model without target dataset is finally evaluated on the corresponding target test set. The results are included in Table 6.10.

setup	test_prs	test_spr
1. KET-PRI		
BASE	0.706	0.648
Rank_WFEDA_without_target_train_data	**0.374 (-47.0%)**	**0.331 (-48.9%)**
2. PET-PRI		
BASE	0.567	0.467
Rank_WFEDA_without_target_train_data	0.523 (-7.8%)	0.424 (-9.2%)
3. FCE-PRI		
BASE	0.359	0.339
Rank_WFEDA_without_target_train_data	0.309 (-13.9%)	0.294 (-13.3%)
4. CAE-PRI		
BASE	0.411	0.410
Rank_WFEDA_without_target_train_data	0.350 (-14.8%)	0.324 (-21.0%)
5. IELTS		
BASE	0.684	0.659
Rank_WFEDA_without_target_train_data	0.586 (-14.3%)	0.558 (-15.3%)
6. FCE-PUB		
BASE	0.662	0.652
Rank_WFEDA_without_target_train_data	0.647 (-2.3%)	0.616 (-5.5%)

Table 6.10: Performance drop on each test set for the ranking W-FEDA model without target dataset in the training set, compared to the baseline model.

First, there is a performance drop for every test set in Table 6.10. For example, although the IELTS exam targets learners from different proficiency levels, without the IELTS dataset in the training set still gives the model a performance drop, which means that the target dataset still plays an important role even if we have multiple source datasets.

We then notice that the relative performance drop for KET-PRI is 47.0% for the Pearson correlation and 48.9% for the Spearman correlation, which are about twice bigger than the CAE-PRI dataset, which has the second biggest drop in terms of model performance. This suggests that transfer learning has limited positive influence if there is a big discrepancy between the data from source and target datasets, which quantitatively supports our hypothesis for the KET exam.

Finally, the performance drop for the FCE-PUB dataset is smaller than the other five datasets. The reason might be that the feature set and model we chose in this thesis were tuned on the FCE-PUB script level, and the model performance was even higher than the human annotators agreement in terms of the Pearson correlation (Section 3.4). The features extracted from other source datasets might still capture useful pattern for predicting FCE-PUB scores. This means that exploring the optimal feature set for each dataset might bring more substantial performance improvement.

6.6 Summary

In this chapter, we studied how to leverage the datasets from different sources to train an automated grader by transfer learning on the target dataset for non-native English learners. The conclusions and contributions we have made include:

- We experimentally demonstrated that including data from other sources can improve model performance.
- We have also compared one-stage and two-stage models in this chapter. We found that two-stage models achieve the best performance in terms of RMSE on five out of six datasets in total, and the new two-stage models we propose in this chapter get the best performance on four out of six development sets in terms of RMSE.
- The best transfer-learning approaches picked from the development set always perform better than BASE, and this improvement is not because of simply changing a different type of model without bringing extra data.

Chapter 7

Jointly Modelling all Meta Properties

In Chapters 4 to 6, we approach ATS from three different types of meta properties including authorship knowledge, inter-exam and intra-exam properties. We notice that there is not a single approach which performs significantly better than the baseline on every dataset, and from Chapter 6, the model performance improvement could also be brought by simply combining different models together. This chapter jointly models all the meta properties we propose in this thesis. We combine the outputs from the three chapters together via ensemble learning, which is a technique to combine multiple learning algorithms together to get a superior predictive performance (Opitz and Maclin, 1999), as the finale of the experiments in this thesis.

7.1 Select Inputs for Ensemble Learning

In this chapter, we combine the models from the previous chapters together via ensemble learning as the finale of the experiments in this thesis. First, we need to choose what outputs from the previous chapters to use to form the inputs of the ensemble learning.

For the authorship knowledge in Chapter 4, we average the predictions from feature fusion with neighbouring text (FF-NT) and concatenated text (FF-CT) (Table 4.3), as for each development set, only one of these two models has the best performance in terms of RMSE.

For the intra-exam properties in Chapter 5, we select the approaches modelling all the intra-exam properties available in each dataset. The intra-exam properties we choose are listed in Table 7.1.

For the inter-exam property in Chapter 6, we choose the transfer-learning setup achieving the best performance on each development set in terms of RMSE (Table 6.7).

We use linear interpolation to average the unrounded outputs we select from each chapter. The interpolation hyper-parameters are tuned on each development set. The outputs of the ensemble learning are then divided by the sum of the interpolation

Dataset	Prompt	Genre	Task
KET-PRI	T	-	-
PET-PRI	T	T	-
FCE-PRI	T	-	T
CAE-PRI	T	-	T
IELTS	T	T	T
FCE-PUB	T	T	T

Table 7.1: Intra-exam properties to conduct W-FEDA for each dataset in this chapter. The performance of W-FEDA on each dataset is available in Table 5.8 and Table 7.2. - means unavailable.

weights and rounded to the nearest valid scores.

7.2 Results

The results of this chapter are reported in Table 7.2. We summarise the meaning of each setup in this table as follows:

- BASE: the support vector regression baseline (SVR) in Chapter 3;
- AUTHORSHIP: the model averaging the outputs of FF-NT and FF-CT from Chapter 4 together;
- INTRA-EXAM: the W-FEDA expanded on the intra-exam properties listed in Table 7.1 from Chapter 5;
- INTER-EXAM: the transfer-learning setup listed in Table 6.9 from Chapter 6;
- EL: the ensemble learning combining results from the previous three chapters.

From Table 7.2, we can see that the ensemble learning (EL) is better than BASE on every metric and dataset. One interesting pattern found in Table 7.2 is that if any method from Chapter 4 to Chapter 6 has a significant improvement on any metric and any dataset, the ensemble-learning method also has a significant improvement on the same metric and dataset. In addition, EL has a significant improvement on the FCE-PUB test set in terms of RMSE, and FCE-PRI on the Pearson and Spearman correlations, and none of the approaches from Chapter 4 to 6 has a significant improvement on these metrics. We suggest that jointly modelling the text properties combines the advantages of the methods from the previous three chapters.

Which properties contribute to the best performance improvement in Table 7.2 heavily depends on which dataset we are looking at. This is similar to what Yannakoudakis (2013) (Section 2.4) and Somasundaran et al. (2014) found that graders trained on

setup	dev_rmse	dev_prs	dev_spr	test_rmse	test_prs	test_spr
1. KET-PRI						
BASE	0.970	0.661	0.585	0.968	0.706	0.648
INTRA_EXAM	0.952	0.678	0.595	**0.904+**	**0.751+**	**0.723+**
INTER_EXAM	0.964	0.664	0.595	0.963	0.713	0.661
EL	**0.943**	**0.683**	**0.603**	0.906+	0.750+	0.719+
2. PET-PRI						
BASE	2.247	0.594	0.541	2.179	0.567	0.467
INTRA_EXAM	2.202	0.609	0.535	2.094+	0.615+	0.510
INTER_EXAM	2.198	0.607	0.532	2.045+	0.637+	**0.557+**
EL	**2.185**	**0.615**	**0.544**	**2.031+**	**0.646+**	0.547+
3. FCE-PRI						
BASE	1.994	0.379	0.363	1.991	0.359	0.339
AUTHORSHIP	1.967	0.398	0.385	1.984	0.343	0.322
INTRA_EXAM	1.954	0.415	0.385	1.953	0.386	0.368
INTER_EXAM	1.938+	0.431+	**0.424+**	1.951+	0.395	0.372
EL	**1.933+**	**0.434+**	0.411+	**1.933+**	**0.402+**	**0.383+**
4. CAE-PRI						
BASE	2.421	0.377	0.341	2.405	0.411	0.410
AUTHORSHIP	2.374+	0.423+	0.395+	2.344+	0.463+	0.462+
INTRA_EXAM	2.379+	0.419+	0.388+	2.358+	0.448	0.446
INTER_EXAM	2.390	0.419+	0.381+	**2.338+**	**0.467+**	**0.468+**
EL	**2.368+**	**0.430+**	**0.398+**	2.348+	0.461+	0.459+
5. IELTS						
BASE	0.701	0.697	0.680	0.693	0.684	0.659
AUTHORSHIP	0.684	0.717	0.699	0.676	0.706	0.696+
INTRA_EXAM	0.699	0.702	0.684	0.684	0.701	0.675
INTER_EXAM	0.663+	0.735+	0.712+	**0.669+**	**0.712+**	**0.702+**
EL	**0.651+**	**0.746+**	**0.718+**	0.671+	0.708+	0.696+
6. FCE-PUB						
BASE	2.402	0.567	0.571	2.569	0.662	0.652
AUTHORSHIP	2.321+	0.600+	**0.607+**	2.516	0.684	0.678
INTRA_EXAM	2.353	0.586	0.584	2.564	0.654	0.636
INTER_EXAM	2.362	0.585	0.577	2.446	0.679	0.661
EL	**2.305+**	**0.605+**	0.603+	**2.419+**	**0.697**	**0.688**

Table 7.2: Results of different setups from Chapter 4 to 6 and ensemble learning (EL) on different datasets. The best setup per dataset is in **bold**. Numbers in green mean improvement and red mean degradation over BASE. + means significantly better ($p < 0.05$) than BASE using the permutation randomisation test (Yeh, 2000) with 2,000 samples.

	Dataset	CEFR	#Words	Author	Intra	Inter	Author+Inter
1	KET-PRI	A2	25-35	–	0.86	0.14	0.14
2	PET-PRI	B1	About 100	–	0.67	0.33	0.33
3	FCE-PRI	B2	140-190	0.55	0.36	0.09	0.64
6	FCE-PUB	B2	120-180	0.36	0.00	0.64	1.00
4	CAE-PRI	C1	220-260	0.82	0.00	0.18	1.00
5	IELTS	ALL	150 for 1st task, 250 for 2nd task	0.30	0.00	0.70	1.00

Table 7.3: Interpolation weights of ensemble learning (Table 7.2) for each dataset. #Words represents the number of words required in each dataset.

different datasets favour different features. We summarise the normalised interpolation hyper-parameters in Table 7.3. Some datasets favour the outputs from the intra-exam properties like KET-PRI and PET-PRI. In contrast, some datasets like CAE-PRI, IELTS and FCE-PUB weigh the outputs with 0.00, although modelling these intra-exam properties in isolation has a performance improvement for these datasets.

The reason for this pattern might be found from the view of Burrows et al. (2015). The authors summarised two types of automated text scoring: automated short answer grading (ASAG) and automated essay scoring (AES). The difference they described is that ASAG focuses on shorter texts and whether any required key points are successfully answered in each text, while AES mainly looks at longer texts and overall writing quality.

In Table 7.3, if we treat modelling authorship knowledge as a pseudo way to increase the training set size, we can then add the weights of Authorship and InterExam together to represent to what extent each dataset prefers bringing more data over distinguishing intra-exam properties. We can see that for the KET-PRI and PET-PRI datasets, the weights of IntraExam are bigger than the weights of InterExam. It means that differentiating intra-exam properties are more important for these two datasets. The reason might be that the length of each text in these two datasets is relatively short, which matches the spirit of ASAG, and differentiating multiple intra-exam properties might be helpful in capturing the key points required for each prompt in ASAG. In comparison, the IntraExam weights for the other four datasets are smaller than the sum of the Authorship and InterExam weights, because these four datasets contain long texts written by learners, which match the spirit of AES. In AES, bringing more data from different sources might help us estimate the overall writing quality of each text more accurately.

Moreover, Burrows et al. (2015) stated that the difference between ASAG and AES is fuzzy. For example, the KET-PRI, PET-PRI and FCE-PRI favour all the meta properties we proposed, which could be related to ASAG and AES. Meanwhile, modelling intra-exam properties in isolation can help improving model performance for the FCE-PRI, CAE-PRI and IETLS datasets in Table 7.2. These results here quantitatively support Burrows et al.'s view.

7.3 Summary

In summary, we combined the outputs from all the previous chapters working on different types of text meta-properties. Explicitly modelling the meta-properties of a text helps us achieve an improvement on all the datasets used in this thesis, compared to modelling no meta-property at all, and each dataset favours different meta-properties differently.

Chapter 8

Conclusions

In summary, we have classified the meta properties of each text into two broad categories: exam-independent and exam-dependent. Exam-dependent properties are further split into inter-exam and intra-exam properties. In this thesis, we have investigated ATS from three different types of meta property. We gave an introduction to the thesis in Chapter 1 and reviewed the related work in Chapter 2. In Chapter 3, we built the baseline ATS model across different datasets, and from Chapter 4 to Chapter 7, we have investigated a number of meta properties not fully studied in previous work:

- In Chapter 4, we investigated authorship knowledge as an example of an exam-independent property. We injected authorship knowledge into ATS by score fusion and feature fusion, and we demonstrated that adding authorship knowledge can improve model performance in most cases. We also chose extra two datasets to demonstrate that this improvement is not merely a result of modelling and remembering the human examiners' bias if this bias exists.

- In Chapter 5, within the same exam, we investigated different intra-exam properties behind a text and how they affect our ATS grader. Three properties were identified including prompt, genre and task. We experimentally showed that modelling properties explicitly and passing them into an ATS grader could be helpful in some cases. We then suggested some reasons why sometimes modelling these properties might not give us a significant improvement or could even lead to performance degradation.

- In Chapter 6, we studied automated assessment on an inter-exam property. We did experiments across different exams, and utilising datasets from other sources can benefit model performance for all the datasets we have. We proposed a new two-stage model, which achieved the best performance on four out of six development sets in terms of root-mean-square error. The transfer-learning model we tuned on each development set always performed better than the baseline on the corresponding test set.

- In Chapter 7, we combined the outputs from Chapter 4 to 6 together by ensemble learning. We showed that this combined model is more consistently better than the baseline model compared to using any single approach from Chapter 4 to 6 in terms of the permutation significant test.

Although this thesis studies the ATS for non-native English speakers on the English exam products from Cambridge Assessment, we think the findings in this thesis could be generalised to other automated assessment applications for different languages, because meta-properties also exist in the texts written for other language exams, which could use the methods we explored in this thesis.

This thesis is only a starting point for studying meta properties in ATS, and we are aware that this thesis has its limitations. Some meta properties are not fully explored, such as formality or sentiment (Section 5.2), and some methods exist in the fields of multi-domain learning (Chapter 5), transfer learning (Chapter 6) and ensemble learning (Chapter 7) which might give us a better performance improvement. We can even design new features for the existence of some meta properties. This thesis is a proof-of-concept to demonstrate whether modelling meta properties can affect ATS, and based on the results from Chapter 4 to Chapter 7, we can conclude that modelling meta properties can improve the performance of an ATS system, although there is not a simple method which works for every dataset.

8.1 Future Work

The future work of this thesis can include:

- In Chapter 4, we can extend our work from two responses to multiple to study further how user behaviour affects grader performance in more detail. For example, Andersen et al. (2013) developed a tutoring software, Cambridge English Write and Improve, which makes it easier to collect more writing from a learner to have a more accurate profile describing the learner.

  In automated speech scoring (such as van Daalen et al. (2015)), it is easy to get multiple responses from the same learner in one exam session, and the outcome of Chapter 4 could be applied to automated speech scoring to study how it differs from ATS in using multiple responses from the same learner.

- In Chapter 5, we can label more intra-exam properties, such as the topic behind each text. Furthermore, Cummins et al. (2016a) and Malinin et al. (2017) explicitly modelled the content of each prompt to improve the performance of prompt relevance detection in learners' responses. Their approaches are a possible direction for our work in order to model to what extent texts from two different prompts/genres/tasks should share their parameters.

- In Chapter 6, we notice that some exams are more similar than others regarding target proficiency levels, and this might also be reflected in the difficulty level of the writing tasks. In the future, we might add proficiency levels into our model to encourage our grader to share more parameters for datasets with similar target proficiency levels.

Appendix A

FCE-PUB Example Prompt

```
<?xml version="1.0" encoding="UTF-8"?><xml><exam m="6" x="0100" y="2000">
Part 1

You must answer this question.

<q n="1">
You recently entered a competition and have just received this letter from the organiser.
Read the letter, on which you have made some notes.
Then, using all the information in your notes, write a suitable reply.

Congratulations!
You have won first prize in our competition --- two weeks at Camp California in the U.S.A.
All accommodation and travel costs are paid for, including transport to and from the airport.
We now need some further information from you:


When would you like to travel?
only July because...

Accommodation at Camp California is in tents or log cabins, which would you prefer?
say which and why

You will have the chance to do two activities while you are at the Camp.
Please choose two from the list below and tell us how good you are at each one.
tell them!

Basketball
Swimming
Golf
```

```
Painting
Climbing
Singing
Sailing
Tennis
Photography
Surfing

Is there anything you would like to ask us?
clothes, money ...?

Yours sincerely
Helen Ryan
Competition Organiser

Write a letter of between 120 and 180 words in an appropriate style on the opposite page.
Do not write any postal addresses.
</q>

Part 2

Write an answer to one of the questions 2 - 5 in this part.
Write your answer in 120 - 180 words in an appropriate style on the opposite page.
Put the question number in the box at the top of page 5.

<q n="2">
Your English class is going to make a short video about daily life at your school.
Your teacher has asked you to write a report, suggesting which lessons and other activities should be filmed, and why.

Write your report.
</q>

<q n="3">
You have recently had a class discussion about shopping.
Now your English teacher has asked you to write a composition, giving your opinions on the following statement:

Shopping is not always enjoyable.

Write your composition.
</q>

<q n="4">
Last month, you enjoyed helping at a pop concert and your pen friend, Kim, wants to
```

```
hear about your experience.
Write a letter to Kim, describing what you did to help and explaining what you particularly liked about the experience.

Write your letter.
Do not write any postal addresses.
</q>

<q n="5">
Answer one of the following two questions based on your reading of one of these set books.
Write (a) or (b) as well as the number 5 in the question box, and the title of the book next to the box.
Your answer must be about one of the books below.

Best Detective Stories of Agatha Christie --- Longman Fiction
The Old Man and the Sea --- Ernest Hemingway
Cry Freedom --- John Briley
Wuthering Heights --- Emily Bront
A Window on the Universe --- Oxford Bookworms Collection

Either
<qq nn="a">
'Sometimes the bad characters in a story are more interesting than the good ones.'
Is this true of the book you have read?
Write a composition, explaining your views with reference to the book or one of the short stories you have read.
</qq>

Or
<qq nn="b">
'This is such a marvellous book you will want to read it again.
Write an article for your college magazine, saying whether you think this statement is true of the book or one of the short stories you have read
</qq>
</q>
</exam></xml>
```

Appendix B

Example Prompts of the FCE-PUB dataset

B.1 First Prompt of 0102_2002_6.xml

You recently had a week's holiday in London.

During your stay you went to the theatre to see a musical show and had a very disappointing evening.

Read the advertisement for the show and the notes you have made. Then, write a letter to the manager of the theatre, explaining what the problems were and asking for some money back.

The Circle Theatre Presents:

OVER THE RAINBOW

London's newest and best musical show

Starring:

Danny Brook and Tina Truelove

no! different actor — disappointing

Times:

14.30 and 19.30

started 20:15!

Tickets:

£10, £15 and £20

Discounts available

Visit our theatre restaurant after the show.

no!

Your perfect evening out!

it wasn't — ask for money back

Write a letter of between 120 and 180 words in an appropriate style on the opposite page.

Do not write any postal addresses.

B.2 First Prompt of 0102_2001_3.xml

Next month, Pete, an English friend who lives in London, will be on holiday in New York for two weeks.

You have agreed to look after his flat for him while he is away.

Read the extract from his letter and your notes.

Then, using all your notes, write a letter to Pete.

I've told my neighbours that you'll come to pick up the keys at 5 o'clock on 4th July.

No!

4 o'clock on 5th

OK?

While you're staying please use anything of mine you need.

Computer, CD player...?

I'm really looking forward to seeing you when I get back on the 20th.

in time for dinner?

To thank you for looking after the flat, I'd like to bring you back something from the USA. Please let me know what you'd like.

Thanks — I'd like...

Write a letter of between 120 and 180 words in an appropriate style on the opposite page.

Do not write any postal addresses.

Bibliography

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. Automatic Text Scoring Using Neural Networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 715–725, Berlin, Germany, August 2016. Association for Computational Linguistics.

Naomi S Altman. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, 46(3):175–185, 1992.

Ilaria L Amerise and Agostino Tarsitano. Correction methods for ties in rank correlations. *Journal of Applied Statistics*, 42(12):2584–2596, 2015.

Øistein E. Andersen. Grammatical error prediction. Technical Report UCAM-CL-TR-794, University of Cambridge, Computer Laboratory, January 2011.

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

Ron Artstein and Massimo Poesio. Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555–596, 2008.

Yigal Attali and Jill Burstein. Automated Essay Scoring With e-rater® v.2.0. *ETS Research Report Series*, 2004(2), 2004.

Margareta Westergren Axelsson. USE - The Uppsala Student English Corpus: An instrument for needs analysis. *ICAME journal*, 24:155–157, 2000.

Marco Baroni, Alessandro Lenci, and Luca Onnis. ISA meets Lara: An incremental word space model for cognitively plausible simulations of semantic learning. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 49–56. Association for Computational Linguistics, 2007.

Yevgeni Berzak, Roi Reichart, and Boris Katz. Reconstructing Native Language Typology from Foreign Language Usage. In *Eighteenth Conference on Computational Natural Language Learning (CoNLL)*, 2014.

Yevgeni Berzak, Roi Reichart, and Boris Katz. Contrastive Analysis with Predictive Power: Typology Driven Estimation of Grammatical Error Distributions in ESL. In *Nineteenth Conference on Computational Natural Language Learning (CoNLL)*, 2015.

Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. Universal Dependencies for Learner English. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 737–746, Berlin, Germany, August 2016. Association for Computational Linguistics.

Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. TOEFL11: A Corpus of Non-Native English. *ETS Research Report Series*, 2013(2), 2013.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

John Blitzer, Ryan McDonald, and Fernando Pereira. Domain Adaptation with Structural Correspondence Learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128. Association for Computational Linguistics, 2006.

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, 2007.

Siegfried Bös and Manfred Opper. Dynamics of batch training in a perceptron. *Journal of Physics A: Mathematical and General*, 31(21):4835, 1998.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

Robert L Brennan and Dale J Prediger. COEFFICIENT KAPPA: SOME USES, MISUSES, AND ALTERNATIVES. *Educational and Psychological Measurement*, 41(3):687–699, 1981.

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

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

Julian Brooke and Graeme Hirst. Measuring Interlanguage: Native Language Identification with L1-influence Metrics. In *LREC*, pages 779–784, 2012.

Christopher Bryant and Mariano Felice. Issues in preprocessing current datasets for grammatical error correction. Technical Report UCAM-CL-TR-894, University of Cambridge, Computer Laboratory, October 2016.

Steven Burrows, Iryna Gurevych, and Benno Stein. The Eras and Trends of Automatic Short Answer Grading. *International Journal of Artificial Intelligence in Education*, 25(1): 60–117, 2015.

Jill Burstein. The E-rater® Scoring Engine: Automated Essay Scoring with Natural Language Processing. In *Automated Essay Scoring: A Cross-Disciplinary Perspective*, pages 113–121. Lawrence Erlbaum Associates Publishers, 2003.

Jill Burstein and Magdalena Wolska. Toward Evaluation of Writing Style: Finding Overly Repetitive Word Use in Student Essays. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 35–42. Association for Computational Linguistics, 2003.

Jill Burstein, Karen Kukich, Susanne Wolff, Chi Lu, Martin Chodorow, Lisa Braden-Harder, and Mary Dee Harris. Automated Scoring Using A Hybrid Feature Identification Technique. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 206–210. Association for Computational Linguistics, 1998.

Jill Burstein, Karen Kukich, Susanne Wolff, Chi Lu, and Martin Chodorow. Enriching Automated Essay Scoring Using Discourse Marking. 2001.

Jill Burstein, Daniel Marcu, and Kevin Knight. Finding the WRITE Stuff: Automatic Identification of Discourse Structure in Student Essays. *IEEE Intelligent Systems*, 18(1): 32–39, 2003.

Jill Burstein, Joel Tetreault, and Slava Andreyev. Using Entity-Based Features to Model Coherence in Student Essays. In *Human language technologies: The 2010 annual conference of the North American chapter of the Association for Computational Linguistics*, pages 681–684. Association for Computational Linguistics, 2010.

Jill Burstein, Joel Tetreault, and Martin Chodorow. Holistic Annotation of Discourse Coherence Quality in Noisy Essay Writing. *Dialogue & Discourse*, 4(2):34–52, 2013.

Andrei M Butnaru and Radu Tudor Ionescu. From Image to Text Classification: A Novel Approach based on Clustering Word Embeddings. *Procedia Computer Science*, 112: 1783–1792, 2017.

Ted Byrt, Janet Bishop, and John B Carlin. BIAS, PREVALENCE AND KAPPA. *Journal of Clinical Epidemiology*, 46(5):423–429, 1993.

Rich Caruana. Multitask Learning. In *LEARNING TO LEARN*, pages 95–133. Springer, 1998.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A Library for Support Vector Machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

Hongbo Chen and Ben He. 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 2013. Association for Computational Linguistics.

Yen-Yu Chen, Chien-Liang Liu, Chia-Hoang Lee, and Tao-Hsing Chang. An Unsupervised Automated Essay-Scoring System. *IEEE Intelligent Systems*, 5(25):61–67, 2010.

Martin Chodorow and Jill Burstein. Beyond Essay Length: Evaluating e-rater's performance on toefl Essays. *ETS Research Report Series*, (1), 2004.

Shamil Chollampatt, Kaveh Taghipour, and Hwee Tou Ng. Neural Network Translation Models for Grammatical Error Correction. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 2016.

Domenic V Cicchetti and Alvan R Feinstein. HIGH AGREEMENT BUT LOW KAPPA: II. RESOLVING THE PARADOXES. *Journal of Clinical Epidemiology*, 43(6):551–558, 1990.

Jacob Cohen. A COEFFICIENT OF AGREEMENT FOR NOMINAL SCALE. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

Jacob Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213, 1968.

David Coniam. Experimenting with a computer essay-scoring program based on ESL student writing scripts. *ReCALL*, 21(2):259–279, 2009.

Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20 (3):273–297, 1995.

Council of Europe. Council for Cultural Co-operation. Education Committee. Modern Languages Division. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Cambridge University Press, 2001. ISBN 9780521803137.

Mădălina Cozma, Andrei M Butnaru, and Radu Tudor Ionescu. Automated essay scoring with string kernels and word embeddings. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages ″503–509″, 2018.

Richard Craggs and Mary McGee Wood. Evaluating Discourse and Dialogue Coding Schemes. *Computational Linguistics*, 31(3):289–296, 2005.

Ronan Cummins, Helen Yannakoudakis, and Ted Briscoe. Unsupervised Modeling of Topical Relevance in L2 Learner Text. In *Proceedings of the Eleventh Workshop on Innovative Use of NLP for Building Educational Applications*, pages 95–104, 2016a.

Ronan Cummins, Meng Zhang, and Ted Briscoe. Constrained Multi-Task Learning for Automated Essay Scoring. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–799, Berlin, Germany, August 2016b. Association for Computational Linguistics.

Daniel Dahlmeier and Hwee Tou Ng. Better Evaluation for Grammatical Error Correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572. Association for Computational Linguistics, 2012.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 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, 2013.

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

Rachele De Felice and Stephen G Pulman. A classifier-based approach to preposition and determiner error correction in L2 English. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 169–176. Association for Computational Linguistics, 2008.

Rachele De Felice and Stephen G Pulman. Automatic Detection of Preposition Errors in Learner Writing. *Calico Journal*, 26(3):512–528, 2009.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by Latent Semantic Analysis. *Journal of the Association for Information Science and Technology (JASIST)*, 41(6):391, 1990.

Janez Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7(Jan):1–30, 2006.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-Task Learning for Multiple Language Translation. 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 1723–1732. Association for Computational Linguistics, 2015.

Fei Dong and Yue Zhang. Automatic Features for Essay Scoring - An Empirical Study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1072–1077, Austin, Texas, November 2016. Association for Computational Linguistics.

Fei Dong, Yue Zhang, and Jie Yang. Attention-based Recurrent Convolutional Neural Network for Automatic Essay Scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162, Vancouver, Canada, August 2017.

Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-Weighted Linear Classification. In *Proceedings of the 25th international conference on Machine learning*, pages 264–271. ACM, 2008.

Mark Dredze, Alex Kulesza, and Koby Crammer. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning*, 79(1-2):123–149, 2010.

Myroslava O Dzikovska, Rodney D Nielsen, and Chris Brew. Towards Effective Tutorial Feedback for Explanation Questions: A Dataset and Baselines. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 200–210. Association for Computational Linguistics, 2012.

Barbara Di Eugenio and Michael Glass. The Kappa statistic: a second look. *Computational Linguistics*, 30(1):95–101, 2004.

Eurostat. Foreign language learning 60% of lower secondary level pupils studied more than one foreign language in 2015 French: second most popular after English. 33, 2017. URL `http://ec.europa.eu/eurostat/documents/2995521/7879483/3-23022017-AP-EN.pdf`.

Theodoros Evgeniou and Massimiliano Pontil. Regularized Multi-Task Learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM, 2004.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9(Aug):1871–1874, 2008.

Youmna Farag, Marek Rei, and Ted Briscoe. An Error-Oriented Approach to Word Embedding Pre-Training. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–158, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

Alvan R Feinstein and Domenic V Cicchetti. HIGH AGREEMENT BUT LOW KAPPA: I. THE PROBLEMS OF TWO PARADOXES. *Journal of Clinical Epidemiology*, 43(6): 543–549, 1990.

Mariano Felice. Artificial error generation for translation-based grammatical error correction. Technical Report UCAM-CL-TR-895, University of Cambridge, Computer Laboratory, October 2016.

Mariano Felice and Ted Briscoe. Towards a standard evaluation method for grammatical error detection and correction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 578–587, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

Mariano Felice and Zheng Yuan. Generating artificial errors for grammatical error correction. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–126. Association for Computational Linguistics, 2014.

Mariano Felice, Christopher Bryant, and Ted Briscoe. Automatic Extraction of Learner Errors in ESL Sentences Using Linguistically Enhanced Alignments. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

Vanessa Wei Feng, Ziheng Lin, and Graeme Hirst. The Impact of Deep Hierarchical Discourse Structures in the Evaluation of Text Coherence. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 940–949, 2014.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 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, 2008.

Angela Ffrench, Graeme Bridges, and Joanna Beresford-Knox. Quality Assurance: A Cambridge ESOL system for managing Writing examiners. In *Research Notes*, volume 49, pages 11–17. University of Cambridge ESOL Examinations, 2012.

Jenny Rose Finkel and Christopher D Manning. Hierarchical Bayesian Domain Adaptation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 602–610. Association for Computational Linguistics, 2009.

Peter W Foltz, Walter Kintsch, and Thomas K Landauer. The Measurement of Textual Coherence with Latent Semantic Analysis. *Discourse Processes*, 25(2-3):285–307, 1998.

Peter W Foltz, Darrell Laham, and Thomas K Landauer. The Intelligent Essay Assessor: Applications to Educational Technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 1(2), 1999.

Jennifer Foster and Øistein E Andersen. GenERRate: generating errors for use in grammatical error detection. In *Proceedings of the fourth workshop on innovative use of nlp for building educational applications*, pages 82–90. Association for Computational Linguistics, 2009.

Roger Garside. The CLAWS word-tagging system. In Roger Garside, Geoffrey Leech, and Geoffrey Sampson, editors, *The Computational Analysis of English: A Corpus-based Approach*. Longman, 1987.

Binyam Gebrekidan Gebre, Marcos Zampieri, Peter Wittenburg, and Tom Heskes. Improving Native Language Identification with TF-IDF Weighting. In *the 8th NAACL Workshop on Innovative Use of NLP for Building Educational Applications (BEA8)*, pages 216–223, 2013.

Debanjan Ghosh, Aquila Khanam, Yubo Han, and Smaranda Muresan. Coarse-grained Argumentation Features for Scoring Persuasive Essays. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 549–554, 2016.

Patrick Gillard and Adam Gadsby. Using a learners corpus in compiling ELT dictionaries. *Learner English on Computer*, pages 159–171, 1998.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *DEEP LEARNING*. MIT Press, 2016.

Patrick Graham and Rodney Jackson. THE ANALYSIS OF ORDINAL AGREEMENT BEYOND WEIGHTED KAPPA. *Journal of Clinical Epidemiology*, 46(9):1055–1062, 1993.

Sylviane Granger. Computer Learner Corpus Research: Current Status and Future Prospects. *Language and Computers*, 52(1):123–145, 2004.

Sylviane Granger, Estelle Dagneaux, Fanny Meunier, and Magali Paquot. *International Corpus of Learner English Version 2*. Presses Universitaires de Louvain, 2002.

Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2. Springer Science & Business Media, 2012.

Kilem Gwet. Kappa Statistic is not Satisfactory for Assessing the Extent of Agreement Between Raters. *Statistical Methods For Inter-Rater Reliability Assessment*, 1(6), 2002.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12(2):115–129, 2006.

Zellig S Harris. DISTRIBUTIONAL STRUCTURE. *Word*, 10(2-3):146–162, 1954.

Michael Heilman and Nitin Madnani. ETS: Domain Adaptation and Stacking for Short Answer Scoring. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 275–279. Association for Computational Linguistics, 2013.

Aurélie Herbelot and Ekaterina Kochmar. 'Calling on the classical phone': a distributional model of adjective-noun errors in learners English. Association of Computational Linguistics, 2016.

Ralf Herbrich, Tom Minka, and Thore Graepel. TrueSkill™: A Bayesian Skill Rating System. In *Advances in neural information processing systems*, pages 569–576, 2007.

Derrick Higgins and Jill Burstein. Sentence similarity measures for essay coherence. In *Proceedings of the 7th International Workshop on Computational Semantics*, pages 1–12, 2007.

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

Sepp Hochreiter and Jürgen Schmidhuber. LONG SHORT-TERM MEMORY. *Neural Computation*, 9(8):1735–1780, 1997.

Andrea Horbach, Dirk Scholten-Akoun, Yuning Ding, and Torsten Zesch. Fine-grained essay scoring of a complex writing task for native speakers. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 357–366, 2017.

Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathiya Keerthi, and Sellamanickam Sundararajan. A Dual Coordinate Descent Method for Large-scale Linear SVM. In *Proceedings of the 25th international conference on Machine learning*, pages 408–415. ACM, 2008.

Radu Tudor Ionescu, Marius Popescu, and Aoife Cahill. Can characters reveal your native language? An independent approach to native language identification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1363–1373. Association for Computational Linguistics, 2014.

Tsunenori Ishioka and Masayuki Kameda. Automated Japanese Essay Scoring System based on Articles Written by Experts. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 233–240. Association for Computational Linguistics, 2006.

Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. A Nested Attention Neural Hybrid Model for Grammatical Error Correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 753–762, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Jing Jiang. A Literature Survey on Domain Adaptation of Statistical Classifiers. 2008. URL `http://sifaka.cs.uiuc.edu/jiang4/domainadaptation/survey`.

Jing Jiang and ChengXiang Zhai. Instance Weighting for Domain Adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Thorsten Joachims. Making Large-Scale SVM Learning Practical. Technical report, Universität Dortmund, June 1998a. SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen.

Thorsten Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Machine Learning: ECML-98*, pages 137–142, 1998b.

Thorsten Joachims. 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, 2002.

Thorsten Joachims. SVMlight: Support Vector Machine. 2008. URL `http://svmlight.joachims.org/`.

Mahesh Joshi, William W Cohen, Mark Dredze, and Carolyn P Rosé. Multi-Domain Learning: When Do Domains Matter? In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1302–1312. Association for Computational Linguistics, 2012.

Mahesh Joshi, Mark Dredze, William W Cohen, and Carolyn Penstein Rosé. What's in a Domain? Multi-Domain Learning for Multi-Attribute Data. In *Proceedings of NAACL-HLT 2013*, pages 685–690, 2013.

William Karush. Minima of Functions of Several Variables with Inequalities as Side Conditions. Master's thesis, University of Chicago, 1939.

Beata Beigman Klebanov, Christian Stab, Jill Burstein, Yi Song, Binod Gyawali, and Iryna Gurevych. Argumentation: Content, Structure, and Relationship with Essay Quality. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 70–75, 2016.

Ekaterina Kochmar. Identification of a Writers Native Language by Error Analysis. Master's thesis, University of Cambridge, 2011.

Ekaterina Kochmar. Error detection in content word combinations. Technical Report UCAM-CL-TR-886, University of Cambridge, Computer Laboratory, May 2016.

Daphne Koller and Nir Friedman. *PROBABILISTIC GRAPHICAL MODELS PRINCIPLES AND TECHNIQUES*. MIT press, 2009.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. Automatically Determining an Anonymous Authors Native Language. In *International Conference on Intelligence and Security Informatics*, pages 209–217. Springer, 2005.

H. W. Kuhn and A. W. Tucker. NONLINEAR PROGRAMMING. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, Calif., 1951. University of California Press.

Karen Kukich. The debate on automated essay grading. *IEEE intelligent systems*, 15(5): 22–27, 2000.

John Lafferty, Andrew McCallum, and Fernando C.N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*, pages 282–289, 2001.

Darrell Laham and Thomas K Landauer. Educational applications of latent semantic analysis. *Measurement and Evaluation in Counseling and Development*, 1998.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. An Introduction to Latent Semantic Analysis. *Discourse Processes*, 25(2-3):259–284, 1998.

Leah S Larkey. 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, 1998.

Claudia Leacock and Martin Chodorow. Automated Grammatical Error Detection. *Automated Essay Scoring: A Cross-Disciplinary Perspective*, pages 195–207, 2003a.

Claudia Leacock and Martin Chodorow. C-rater: Automated Scoring of Short-Answer Questions. *Computers and the Humanities*, 37(4):389–405, 2003b.

Ching-Pei Lee and Chih-Jen Lin. Large-Scale Linear RankSVM. *Neural Computation*, 26 (4):781–817, 2014.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. Automatically Evaluating Text Coherence Using Discourse Relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 997–1006. Association for Computational Linguistics, 2011.

Deryle Lonsdale and Diane Strong-Krause. Automated Rating of ESL Essays. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing - Volume 2*, pages 61–67. Association for Computational Linguistics, 2003.

Wei Lu, Hai Leong Chieu, and Jonathan Löfgren. A General Regularization Framework for Domain Adaptation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 950–954. Association for Computational Linguistics, 2016.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. MULTI-TASK SEQUENCE TO SEQUENCE LEARNING. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

Andrey Malinin, Kate Knill, and Mark JF Gales. A HIERARCHICAL ATTENTION BASED MODEL FOR OFF-TOPIC SPONTANEOUS SPOKEN RESPONSE DETECTION. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 397–403. IEEE, 2017.

Melvin Earl Maron. Automatic Indexing: An Experimental Inquiry. *Journal of the ACM (JACM)*, 8(3):404–417, 1961.

Andrew McCallum and Kamal Nigam. A Comparison of Event Models for Naive Bayes Text Classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Madison, WI, 1998.

John CP Milton and Nandini Chowdhury. Tagging the interlanguage of Chinese learners of English. In *Proceedings joint seminar on corpus linguistics and lexicology, Guangzhou and Hong Kong (Language Centre, HKUST, Hong Kong, 1994)*, 1994.

Eleni Miltsakaki and Karen Kukich. Automated Evaluation of Coherence in Student Essays. In *Proceedings of LREC 2000*, 2000.

Eleni Miltsakaki and Karen Kukich. Evaluation of text coherence for electronic essay scoring systems. *Natural Language Engineering*, 10(1):25–55, 2004.

Tom Mitchell, Terry Russell, Peter Broomhead, and Nicola Aldridge. Towards robust computerised marking of free-text responses. Technical report, Loughborough University, 2002.

Tom M. Mitchell. *MACHINE LEARNING*. McGraw-Hill, 1997.

Michael Mohler, Razvan Bunescu, and Rada Mihalcea. Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency Graph Alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 752–762. Association for Computational Linguistics, 2011.

Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. GLEU Without Tuning. *arXiv preprint arXiv:1605.02592*, 2016.

Nadja Nesselhauf. Learner corpora and their potential for language teaching. In John McH. Sinclair, editor, *How to Use Corpora in Language Teaching*, volume 12 of *Studies in Corpus Linguistics*, pages 125–156. 2004.

Andrew Ng. CS229 Lecture notes. URL `http://cs229.stanford.edu/notes/`.

Diane Nicholls. 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, 2003.

Rodney D Nielsen, Wayne Ward, and James H Martin. Classification Errors in a Domain-Independent Assessment System. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 10–18. Association for Computational Linguistics, 2008.

David Opitz and Richard Maclin. Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

Robert Östling, Andre Smolentzov, Björn Tyrefors Hinnerich, and Erik Höglin. Automated Essay Scoring for Swedish. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 42–47, 2013.

Ulrike Pado. Question Difficulty – How to Estimate Without Norming, How to Use for Automated Grading. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–10, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

Ellis B Page. THE USE OF THE COMPUTER IN ANALYZING STUDENT ESSAYS. *International Review of Education*, 14(2):210–225, 1968.

Ellis B Page and Nancy S Petersen. The Computer Moves into Essay Grading: Updating the Ancient Test. *Phi Delta Kappan*, 76(7):561, 1995.

Ellis Batten Page. Computer Grading of Student Prose, Using Modern Concepts and Software. *The Journal of Experimental Education*, 62(2):127–142, 1994.

Ellis Batten Page. Project Essay Grade: PEG. *Automated Essay Scoring: A Cross-Disciplinary Perspective*, pages 43–54, 2003.

Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Cross-Domain Sentiment Classification via Spectral Feature Alignment. In *Proceedings of the 19th International Conference on World Wide Web*, pages 751–760. ACM, 2010.

Karl Pearson. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895.

Diana Pérez-Marín, Ismael Pascual-Nieto, and Pilar Rodríguez. Computer-assisted assessment of free-text answers. *The Knowledge Engineering Review*, 24(4):353–374, 2009.

Isaac Persing and Vincent Ng. Modeling Thesis Clarity in Student Essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 260–269, 2013.

Isaac Persing and Vincent Ng. Modeling Prompt Adherence in Student Essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1534–1543, 2014.

Isaac Persing and Vincent Ng. 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, 2015.

Isaac Persing, Alan Davis, and Vincent Ng. Modeling Organization in Student Essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 229–239. Association for Computational Linguistics, 2010.

Peter Phandi, Kian Ming A. Chai, and Hwee Tou Ng. 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 2015. Association for Computational Linguistics.

John C. Platt. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. *Technical Report MSR-TR-98-14*, April 1998.

Robert Gilmore Pontius Jr and Marco Millones. Death to Kappa: birth of quantity disagreement and allocation disagreement for accuracy assessment. *International Journal of Remote Sensing*, 32(15):4407–4429, 2011.

David MW Powers. The Problem with Kappa. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 345–355. Association for Computational Linguistics, 2012.

Norma A Pravec. Survey of learner corpora. *ICAME journal*, 26(1):8–14, 2002.

Stephen G Pulman and Jana Z Sukkarieh. Automatic Short Answer Marking. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, pages 9–16. Association for Computational Linguistics, 2005.

Chaitanya Ramineni, Catherine S Trapani, David M Williamson, Tim Davey, and Brent Bridgeman. Evaluation of the e-rater® Scoring Engine for the GRE® Issue and Argument Prompts. *ETS Research Report Series*, 2012(1), 2012.

Bob Rehder, M.E. Schreiner, Michael B.W. Wolfe, Darrell Laham, Thomas K Landauer, and Walter Kintsch. Using Latent Semantic Analysis to Assess Knowledge: Some Technical Considerations. *Discourse Processes*, 25(2-3):337–354, 1998.

Marek Rei. Detecting Off-topic Responses to Visual Prompts. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 188–197, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

Marek Rei, Gamal Crichton, and Sampo Pyysalo. Attending to Characters in Neural Sequence Labeling Models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 309–318, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

Carolyn P Rosé, Antonio Roque, Dumisizwe Bhembe, and Kurt Vanlehn. A Hybrid Text Classification Approach for Analysis of Student Essays. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing-Volume 2*, pages 68–75. Association for Computational Linguistics, 2003.

Frank Rosenblatt. THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN. *Psychological review*, 65(6): 386, 1958.

Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To Transfer or Not To Transfer. In *NIPS 2005 workshop on transfer learning*, volume 898, pages 1–4, 2005.

Lawrence M. Rudner and Tahung Liang. Automated Essay Scoring Using Bayes Theorem. *The Journal of Technology, Learning and Assessment*, 1(2), 2002.

Fei Sha and Fernando Pereira. Shallow Parsing with Conditional Random Fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 134–141. Association for Computational Linguistics, 2003.

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

Mark D Shermis and Jill C Burstein. *Automated Essay Scoring: A Cross-disciplinary Perspective*. Routledge, 2003.

Mark D Shermis, Howard R Mzumara, Jennifer Olson, and Susanmarie Harrington. On-line Grading of Student Essays: PEG goes on the World Wide Web. *Assessment & Evaluation in Higher Education*, 26(3):247–259, 2001.

Julius Sim and Chris C Wright. The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements. *Physical Therapy*, 85(3):257–268, 2005.

Morton Slater. LAGRANGE MULTIPLIERS REVISITED. Technical report, Cowles Foundation for Research in Economics, Yale University, 1959.

Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

Swapna Somasundaran, Jill Burstein, and Martin Chodorow. 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 2014. Dublin City University and Association for Computational Linguistics.

Swapna Somasundaran, Chong Min Lee, Martin Chodorow, and Xinhao Wang. Automated Scoring of Picture-based Story Narration. In *Proceedings of the tenth workshop on Innovative Use of NLP for Building Educational Applications*, pages 42–48, 2015.

Swapna Somasundaran, Brian Riordan, Binod Gyawali, and Su-Youn Yoon. Evaluating Argumentative and Narrative Essays using Graphs. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1568–1578, 2016.

Swapna Somasundaran, Michael Flor, Martin Chodorow, Hillary Molloy, Binod Gyawali, and Laura McCulla. Towards Evaluating Narrative Quality In Student Writing. *Transactions of the Association for Computational Linguistics*, 6:91–106, 2018.

Wei Song, Tong Liu, Ruiji Fu, Lizhen Liu, Hanshi Wang, and Ting Liu. Learning to Identify Sentence Parallelism in Student Essays. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 794–803, 2016.

Wei Song, Dong Wang, Ruiji Fu, Lizhen Liu, Ting Liu, and Guoping Hu. Discourse Mode Identification in Essays. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 112–122, 2017.

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

Charles Spearman. THE PROOF AND MEASUREMENT OF ASSOCIATION BETWEEN TWO THINGS. *American Journal of Psychology*, 15(1):72–101, 1904.

Jana Z Sukkarieh, Stephen G Pulman, and Nicholas Raikes. Auto-marking: using computational linguistics to score short, free-text responses. In *Proceedings of 29th International Association for Educational Assessment (IAEA) Annual Conference*, 2003.

Raymond Hendy Susanto, Peter Phandi, and Hwee Tou Ng. Systems Combination for Grammatical Error Correction. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 951–962, 2015.

Anaïs Tack, Thomas François, Sophie Roekhaut, and Cédrick Fairon. Human and Automated CEFR-based Grading of Short Answers. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 169–179, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

Kaveh Taghipour and Hwee Tou Ng. A Neural Approach to Automated Essay Scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas, November 2016. Association for Computational Linguistics.

Yi Tay, Minh C Phan, Luu Anh Tuan, and Siu Cheung Hui. SKIPFLOW: Incorporating Neural Coherence Features for End-to-End Automatic Text Scoring. In *Proceedings of AAAI 2018*, 2018.

Joel Tetreault, Daniel Blanchard, and Aoife Cahill. A Report on the First Native Language Identification Shared Task. In *Proceedings of the eighth workshop on innovative use of NLP for building educational applications*, pages 48–57, 2013.

Joel R Tetreault and Martin Chodorow. The Ups and Downs of Preposition Error Detection in ESL Writing. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 865–872. Association for Computational Linguistics, 2008.

Joel R Tetreault, Daniel Blanchard, Aoife Cahill, and Martin Chodorow. Native Tongues, Lost and Found: Resources and Empirical Evaluations in Native Language Identification. In *Proceedings of COLING 2012*, pages 2585–2602, 2012.

John S Uebersax. Diversity of Decision-Making Models and the Measurement of Interrater Agreement. *Psychological Bulletin*, 101(1):140, 1987.

Salvatore Valenti, Francesca Neri, and Alessandro Cucchiarelli. An Overview of Current Research on Automated Essay Grading. *Journal of Information Technology Education: Research*, 2(1):319–330, 2003.

Roger van Daalen, Katherine Mary Knill, and Mark John Gales. Automatically Grading Learners English Using a Gaussian Process. ISCA, 2015.

Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer Science+Business Media, 1999.

Henning Wachsmuth, Nona Naderi, Ivan Habernal, Yufang Hou, Graeme Hirst, Iryna Gurevych, and Benno Stein. Argumentation Quality Assessment: Theory vs. Practice. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 250–255, 2017.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A Survey on Transfer Learning. *Journal of Big Data*, 3(1):9, 2016.

Frank Wilcoxon. INDIVIDUAL COMPARISONS BY RANKING METHODS. *Biometrics Bulletin*, 1(6):80–83, 1945.

Caroline Williams. The Cambridge Learner Corpus for researchers on the English Profile Project Version 2. 2008.

David M Williamson. A Framework for Implementing Automated Scoring. In *Annual Meeting of the American Educational Research Association and the National Council on Measurement in Education, San Diego, CA*, 2009.

Pengcheng Yang, Xu Sun, Wei Li, and Shuming Ma. Automatic Academic Paper Rating Based on Modularized Hierarchical Convolutional Neural Network. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, 2: 496–502, 2018.

Helen Yannakoudakis. Automated assessment of English-learner writing. Technical Report UCAM-CL-TR-842, University of Cambridge, Computer Laboratory, October 2013.

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

Helen Yannakoudakis and Ronan Cummins. Evaluating the performance of Automated Text Scoring systems. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–223, 2015.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 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 2011. Association for Computational Linguistics.

Alexander Yeh. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 947–953. Association for Computational Linguistics, 2000.

Attali Yigal and Jill Burstein. Automated Essay Scoring With E-rater v.2.0. *Journal of Technology, Learning, and Assessment,(JTLA)*, 4(3), 2006.

Su-Youn Yoon and Derrick Higgins. Non-English Response Detection Method for Automated Proficiency Scoring System. In *Proceedings of the 6th workshop on Innovative Use of NLP for Building Educational Applications*, pages 161–169. Association for Computational Linguistics, 2011.

Zheng Yuan. Grammatical error correction in non-native English. Technical Report UCAM-CL-TR-904, University of Cambridge, Computer Laboratory, March 2017.

Zheng Yuan, Ted Briscoe, and Mariano Felice. Candidate re-ranking for SMT-based grammatical error correction. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 256–266, San Diego, CA, June 2016. Association for Computational Linguistics.

Torsten Zesch, Michael Wojatzki, and Dirk Scholten-Akoun. Task-Independent Features for Automated Essay Grading. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 224–232, 2015.

Meng Zhang, Xie Chen, Ronan Cummins, Øistein Andersen, and Ted Briscoe. The Effect of Adding Authorship Knowledge in Automated Text Scoring. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*, New Orleans, USA, June 2018. Association for Computational Linguistics.