UNIVERSITY OF
CAMBRIDGE
Department of Computer
Science and Technology

Multi-Task Learning for Multi-Dimensional Essay Scoring

Gabrielle Gaudeau

Wolfson College

June 2023

Submitted in partial fulfillment of the requirements for the
Master of Philosophy in Advanced Computer Science

Total page count: 75

Main chapters (excluding front-matter, references and appendix): 47 pages (pp 8–54)

Main chapters word count: 14,933

```
File: report.tex
Encoding: utf8
Sum count: 14933
Words in text: 13225
Words in headers: 105
Words outside text (captions, etc.): 1528
Number of headers: 52
Number of floats/tables/figures: 33
Number of math inlines: 70
Number of math displayed: 5
Subcounts:
  text+headers+captions (#headers/#floats/#inlines/#displayed)
  0+6+0 (1/0/0/0) _top_
  923+7+61 (4/0/0/0) Chapter: Introduction
  3273+18+347 (13/3/42/5) Chapter: Background
  1180+16+70 (7/2/0/0) Chapter: Related Work
  3541+22+540 (13/15/20/0) Chapter: Methodology
  3309+32+484 (11/13/8/0) Chapter: Evaluation
  999+4+26 (3/0/0/0) Chapter: Conclusion
```

Methodology used to generate that word count:

```
\newcommand{\detailtexcount}[1]{%
    \immediate\write18{
        texcount -merge -sum -q -sub=chapter #1.tex > #1.wcdetail
    }%
    \verbatiminput{#1.wcdetail}%
}
```

And %TC:ignore/%TC:endignore tags were place around sections that do not contribute to the word count (declaration, abstract, acknowledgements, front matter, bibliography and appendices).

Declaration

I, Gabrielle Gaudeau of Wolfson College, being a candidate for the Master of Philosophy in Advanced Computer Science, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

Signed: Gabrielle Gaudeau

Date: 01/06/2023

Abstract

The Automated Assessment (AA) community, specialising in the automatic evaluation of essays, has seen a recent rise in transformer-based models, popularised by BERT. These neural-based models often outperform traditional feature-based approaches and obviate the need for manual feature engineering. However, their need for large amounts of annotated data is a serious bottleneck for the field.

While essay quality dimensions of syntax and grammar have been extensively studied, other core aspects of writing, like cohesion, present a much bigger challenge. In this study, we explore multi-task learning (MTL) as a possible novel solution to this imbalance and build a neural-based system capable of scoring student essays along six different dimensions of increasing complexity using the ELLIPSE dataset. These are conventions, grammar, syntax, phraseology, vocabulary, and cohesion.

Our main finding is that MTL can in fact help improve the predictions for our considered range of essay quality dimensions, and does so even for the most challenging amongst them. We hope that, in turn, these results will lay the foundations for future AA systems capable of providing comprehensive, multi-dimensional feedback to students and teachers on essays, where prior work primarily focused on producing a single holistic score.

Acknowledgements

This project would not have been possible without the guidance of my supervisors, Øistein Andersen and Zheng Yuan. I am extremely grateful and humbled to have had so much of their time and knowledge at my disposal. I would also like to thank the Cambridge University Institute for Automated Language Teaching and Assessment (ALTA)[1] for granting me access to their computing resources and their continued support.

I also want to give special thanks to The Learning Agency Lab[1] who co-created the ELLIPSE dataset which was central to this study, and more particularly to Natalie Rambis, Perpetual Baffour, and Scott Crossley, who most kindly and eagerly agreed to us using and including the dataset's annotation guidelines in this report though they had not yet been made public, and shared with us a soon-to-be-released paper describing the corpus creation in much more detail.

Finally, I would be remiss to fail to mention my family and friends whose unwavering support was most precious during this incredibly stressful year. Above all, I thank my partner for his unfailing kindness and understanding throughout this project. I look forward to where the future may take us.

[1] For more information, visit http://alta.cambridgeenglish.org.

Contents

Chapter 1

Introduction

Technology has altered the way in which we interact with people and placed a greater emphasis on written communication via text messages, emails, social-media posts, etc. This change is driven by the rapid development of intelligent text entry systems and writing-aid tools. From the keyboards of our smartphones (e.g., SwiftKey[1]), to our email (e.g., GMail[2]) and text editors (e.g., Overleaf[3]), they are everywhere, and seem to do everything: from simple grammar-checking, and predictive suggestions of words and phrases, to large-scale human-like text generation (e.g., OpenAI's ChatGPT[4]).

The ubiquity of these systems is changing our relationship to writing, and will continue to do so for the coming years, in ways we have yet to fully understand (Abbasi, 2020; Arnold et al., 2020). Yet, writing remains a fundamentally human skill (Wen and Walters, 2022), one that we must learn to enhance our academic and professional prospects (Arcon et al., 2017). Until proven otherwise, technology will not replace the need for humans to be proficient in writing. Instead, *could we use machines to help us learn to write?*

1.1 Context

Natural Language Processing (NLP) is the field which sits at the crossroads between **Machine Learning (ML)** and Linguistics and aims to help computer systems understand and manipulate language (Chowdhary, 2020). It has many uses, including the ones we mentioned above, but if we set our focus on education, perhaps its most important application is **Automated Assessment (AA)** (Ke and Ng, 2019), a field which pushes the limits of machine-assisted learning a little further every day.

[1] See https://www.microsoft.com/en-us/swiftkey.

[2] The release of the "Help Me Write" feature was announced by Google (2023) on the 10$^{\text{th}}$ May, just a few years after SmartCompose (Chen et al., 2019).

[3] The very software used to write this report comes with simple spell-checking capabilities. For more information, visit https://www.overleaf.com/learn.

[4] You can try ChatGPT via https://openai.com/blog/chatgpt.

AA consists in the automatic evaluation of human writing (Mayfield and Black, 2020). Originally used to alleviate the marking load of standardised tests such as TOEFL and GMAT (Chodorow and Burstein, 2004; Chen et al., 2016), past AA work primarily focused on holistic scoring: summarising the quality of an essay with a single score (Phillips, 2007). More recently, AA research is turning to multi-dimensional essay scoring (Higgins et al., 2004; Louis and Higgins, 2010; Somasundaran et al., 2014; Persing and Ng, 2014; Kaneko et al., 2020): breaking down single holistic scores into several essay quality dimension scores (coherence, syntax, relevance to prompt, etc.) to better highlight the strengths and weaknesses of a student's writing (Ke and Ng, 2019). This switch is encouraging the emergence of automatic systems which provide richer essay evaluations (Burstein et al., 2004) which are slowly making their way into the classrooms where quick personalised formative feedback is particularly valued (Wilson and Roscoe, 2020; Li et al., 2014).

Traditionally, research in AA prioritised simple feature-based approaches, but with the recent surge of interest in the transformer architecture (Vaswani et al., 2017), neural networks have gained favour (Taghipour and Ng, 2016; Alikaniotis et al., 2016; Mayfield and Black, 2020). These perform on par with feature-based systems, and eliminate the need for expensive feature engineering. This gain comes at the cost of needing increasingly large quantities of annotated data for training and an inherent lack of interpretability of the models and their results (Hall et al., 2017; Du et al., 2019).

Unfortunately, the different dimensions of essay quality are not equally studied or simple to evaluate (Ke and Ng, 2019). While the detection of grammatical and mechanical errors has been extensively and successfully explored (Chen et al., 2020), dimensions of coherence (Higgins et al., 2004), thesis clarity (Persing and Ng, 2013), and persuasiveness (Stab and Gurevych, 2014) remain challenging discourse-level problems to this day which require deep linguistic understanding capabilities, far surpassing those of current state-of-the-art essay scoring systems (Ke and Ng, 2019).

1.2 Approach and Contributions

To tackle the imbalance between essay quality dimensions, we propose to explore **multi-task learning (MTL)**. Inspired by how humans generalise situation-specific knowledge to similar tasks (Ruder, 2017), MTL allows neural models to learn from multiple objectives, leveraging information from related tasks to improve performance on tasks which are considered harder or for which the data is limited (Caruana, 1993; Andersen et al., 2021). In this study, we seek to investigate whether the theoretical merits of MTL show, in practice, actual promise for multi-dimensional essay scoring.

We use ELLIPSE (Crossley et al., forthcoming), a corpus of argumentative essays written by 8^{th} to 12^{th} grade English language learners (ELLs) (Arcon et al., 2017) in the United States and scored according to six dimensions of essay composition (Cohesion, Syntax,

Vocabulary, Phraseology, Grammar, and Conventions), which was published in the English Language Learning Feedback Prize Kaggle competition[5] over six months ago. Since this dataset is so young, no academic work has yet been published using it. Hence, our first step is to establish a simple and reproducible baseline model trained, fine-tuned and tested on ELLIPSE, which we make available to everyone who might pursue work on this dataset in the future. Then, building on top of the baseline architecture, we design an MTL neural system capable of scoring essays along the six dimensions simultaneously.

We evaluated this model against our baseline using the Pearson and Spearman rank correlation metrics, and the Root Mean Square Error (RMSE), three performance measures which are standard in the field of AA (Yannakoudakis et al., 2011; Ke and Ng, 2019), and found that the MTL approach can in fact improve on the baseline, and, more importantly, does so even for some of the trickiest of dimensions. Given the promise of these results, we begin to motivate future work in MTL for multi-dimensional essay scoring.

1.3 Report Structure

Our project report is structured as follows:

Chapter 2: The Background chapter starts by briefly describing the task of automated essay scoring and continues with an extensive coverage of the knowledge required to understand the rest of the report.

Chapter 3: Next, we introduce the different bodies of research and key papers upon which this study rests. Additionally, we give an overview of the common evaluation strategies used for automated essay scoring systems which we will employ in the evaluation of our models.

Chapter 4: In Methodology, we present our step-by-step approach to building an MTL multi-dimensional essay scoring model starting from a simple holistic scoring baseline. Here, the datasets and implementation details are carefully described.

Chapter 5: This chapter is dedicated to the evaluation of our MTL model results. We begin to analyse the potential benefits of the proposed approach and then test our hypotheses in further experiments, and finally discuss the limitations of our study.

Chapter 6: The Conclusion chapter summarises the study and its contributions, and includes some ideas for future research.

[5] See https://www.kaggle.com/competitions/feedback-prize-english-language-learning/data.

Chapter 2

Background

This chapter lays the foundations of this project, introducing key Machine Learning (ML) notions, and Automated Assessment (AA)-specific architectures, and metrics.

2.1 Learning

More and more we find ourselves surrounded by so-called *intelligent* systems which help us in our everyday tasks (Markauskaite et al., 2022). Properly understanding how these work is critical to building useful mental models (Lin et al., 2020) but, for many, the question remains: what do we mean by machines that *learn*? "Can machines think?" (Turing, 1950). Concretely, Mitchell (1997) offers the following definition:

> **Definition 2.1.1** (Machine Learning)**.** A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

From the perspective of this project, we require a computer program, or **model**, to be capable of learning how to automatically mark a written essay (T). That is, it should be able to improve its ability to predict an essay's true score, measured according to some metric (P), by gaining experience from viewing correctly annotated essays (E). This essay score can either be an overall grade or a mark for a particular dimension (e.g., cohesion).

The process of learning through experience is called **training**. In our case, provided a set of correct essay–score pairs as input, the computer program tries to approximate the function, or **hypothesis**, which best describes the relationship between the input features (written essays) and the target values (correct essay scores). We denote **training set** the labelled set of data provided to the program for experience gain during training.

This particular setting is called **supervised learning** and differs from semi-supervised, unsupervised or reinforcement learning methods which do not (solely) rely on annotated training data to learn. Whichever form of learning we choose, the model can, once trained, be applied to unseen essays whose scores are unknown to the model during **testing**. This is called the **test set**, and our system's performance can be measured by comparing the automatically predicted scores it outputs for that set to the scores awarded by human markers on the same essays. We call the latter the **gold standard** (Williamson et al., 2012b). Since virtually all state-of-the-art AA systems are supervised, it is also how we will approach our task. For the application of other learning approaches within the field, refer to Chen et al. (2010) and Wang et al. (2018).

In the next section, we introduce an off-the-shelf supervised learning algorithm typically used for training AA models: **regression**.

2.2 Regression

So far, we have described our task as automatically predicting essay scores, which are, in effect, real-valued variables. This is called a regression task and differs from **classification** which assigns discrete labels (e.g., CEFR levels[1]) instead. We will be focusing on regression in this study but classification tasks do in fact exist within the field of AA. See Rudner and Liang (2002), Farra et al. (2015), Vajjala (2017), and Nguyen and Litman (2018) for related work.

There exists several different supervised learning algorithms for regression: (1) **linear regression** (Page, 1966; Landauer et al., 2003; Miltsakaki, 2004; Attali and Burstein, 2006; Beigman Klebanov et al., 2013; Faulkner, 2014; Crossley et al., 2015; Beigman Klebanov et al., 2016), (2) **support vector regression** (Persing et al., 2010; Persing and Ng, 2013, 2014, 2015; Cozma et al., 2018), and (3) **sequential minimal optimisation (SMO)** (Vajjala, 2017), have notably been used for automated essay scoring. Here, we introduce only (1) which is defined as the task of learning the best **linear hypothesis function** to describe our training data. This function can be either univariate in its simplest form (one input feature), or multivariate.

In practice, suppose that for each written essay in the training set, we obtain a single real-valued input feature vector through some process (more on this in Section 2.4). Given such a vector $\mathbf{x}$, we define the linear hypothesis function $h_{\mathbf{w}}$ as:

$$h_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + \cdots + w_N x_N, \tag{2.1}$$

[1] Common European Framework of Reference for Languages (North and Piccardo, 2020) levels correspond to language proficiency levels ranging from A1 (elementary) to C2 (complete proficiency) from a second-language learner's perspective.

which also happens to be the definition of a linear function in N variables (Jacob, 1995). Here, N is the size of the training set, and the real-valued coefficients of $\mathbf{w} = [w_0, \cdots, w_N]$ are parameters which essentially define the hypothesis function. These are the **weights**, and w_0 is the **bias** term,[2] and they need to be learned to obtain the best possible fit to the training data, or equivalently, to minimises some sort of **loss** or **cost** (a penalty resulting from a bad prediction) on the training set. Only then can we predict y, the score of the original written essay corresponding to $\mathbf{x}$, as accurately as possible. For this, we can use the **Least Squared Error (LSE)** function summed over all training examples and take the square root:

$$\text{Cost}(h_{\mathbf{w}}) = \sum_{i=1}^{N} \text{LSE}(y_i, h_{\mathbf{w}}(\mathbf{x}_i)) = \frac{1}{2N} \sum_{i=1}^{N} (y_i - h_{\mathbf{w}}(\mathbf{x}_i))^2, \tag{2.2}$$

where for all $1 \leq i \leq N$, we denote $\mathbf{x}_i$ the feature input vector of the i-th training example and y_i its corresponding predicted score. This cost function is also known as the **Mean Squared Error (MSE)** and amounts to solving the following optimisation problem:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\text{argmin}} \sum_{i=1}^{N} \text{Cost}(y_i, h_{\mathbf{w}}), \tag{2.3}$$

where $\mathbf{w}^*$ corresponds to the best vector of weights.

Gradient descent is a popular optimisation algorithm which can be used to solve this.[3] We start by choosing any point $\mathbf{w}$ in the weight space at random.[4] Then, we simultaneously update each individual weight until we converge on the minimum possible loss, using the following weight-updating formula:

$$w_j \leftarrow w_j - \alpha \frac{1}{N} \sum_{i=1}^{N} x_{i,j} (y_i - h_{\mathbf{w}}(\mathbf{x_i})) \tag{2.4}$$

where α is called the **learning rate**, and $x_{i,j}$ is the j-th element of the i-th example in the training set. There exists a multitude of different ways of evaluating linear regression model performance. Taking the square root of Equation (2.2) yields another popular cost function called the **Root Mean Squared Error (RMSE)** (Karunasingha, 2022), which we will meet again in later chapters.

[2] Adding a bias weight, independent of any of the input features, ensures that the hypothesis function can be fitted to data that does not pass through the origin. This term is omnipresent in ML but adds little to what we wish to convey here. Hence we will, for the most part, ignore it.

[3] There exists other, faster optimisation techniques but they are beyond the scope of this study. Further, this optimisation problem can also be solved analytically (Russell and Norvig, 2003, Sections 18.6.1 & 18.6.2).

[4] Random initialisations of parameters are frequent in ML techniques and responsible for the stochasticity of the models we will be using. It is the reason why we will need to fix a random seed value in our experiments (Section 4.1.4).

In regression, we distinguish three parts: a parameterised model, some data and an optimisation strategy (a way to find the optimal weights). We will continue to see this triptych as we move on to more advanced methods. Next, we take a look at the basic **feed-forward neural network**, a natural extension of linear regression.

2.3 Artificial Neural Networks

The **artificial neural network** is one of the oldest ML techniques (Pomerleau, 1988), dating back to the late 1940s, shortly after World War II. Inspired by the neurosciences of the time, early models drew from nature (McCulloch and Pitts, 1943): modelling thinking and learning as electrochemical signals propagated through a network of brain cells (neurons). Those who sought faithful and realistic representations became the pioneers of modern computational neuroscience (Russell and Norvig, 2003). Others turned their attention to the abstract properties of neural networks: their ability to tolerate noise, perform parallel distributed processing, and learn (Rosenblatt, 1958), giving birth to **Artificial Intelligence (AI)**.

NLP was born out of AI from a desire to grant machines the ability to understand and interpret human language (Chowdhury, 2005). The field has evolved massively and neural-based models have achieved stunning results in various NLP tasks, including AA (Ke and Ng, 2019), the focus of this study.

2.3.1 Definition

An artificial neural network is, in essence, a **weighted graph** (Diestel, 2017, Section 1.1) whose edges may be undirected, but are more commonly directed forward as in Figure 2.1. It is formed of an input **layer**, a number of intermediate hidden layers, and an output layer. Each is composed of a certain number of nodes (also called **units** or **neurons**) and their associated **activation functions** (Section 2.3.2). All units can be interconnected to every other node of the direct neighbouring layers and a numeric weight is associated to every one of these links determining the strength and sign of the connection. These can be collected to form a **weight matrix** $\mathbf{W}^{(l)}$ within any single layer l. Together, the weight matrices of a neural network are the parameters we will want to learn during training.

Neural networks and linear regression (Section 2.2) share much of the same mathematics: we can think of neural networks as hypothesis functions $h_{\mathcal{W}}$ parameterized by the set of their weight matrices $\mathcal{W}$. They are, however, more powerful. A minimal neural network, of at least one single hidden layer, can learn any function, including non-linearities (Jurafsky and Martin, 2021, p.134). This is not the case of linear regression, which assumes linearity. Multi-layer networks can also represent learning problems with multiple outputs. In such cases, we should think of the networks as implementing a vector function $\mathbf{h}_{\mathcal{W}}$ with a target output vector $\mathbf{y}$ rather than a scalar function $h_{\mathcal{W}}$ and scalar prediction y.

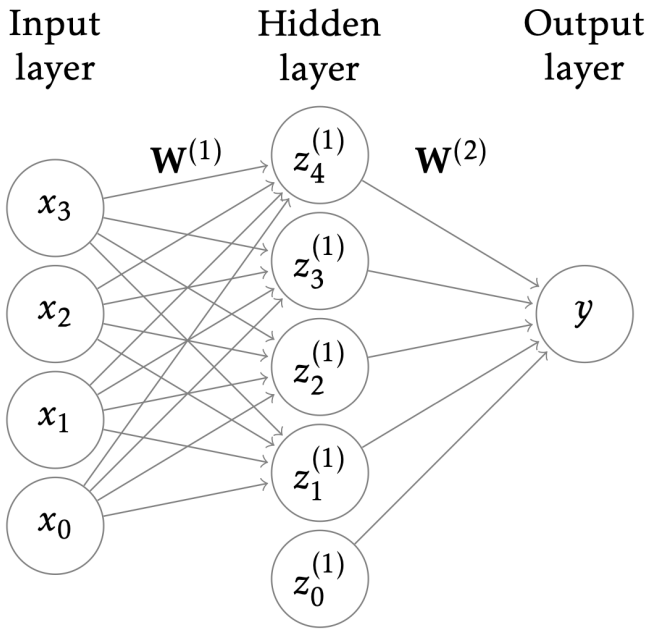

Figure 2.1: A simple feed-forward fully-connected[5] neural network architecture consisting of an input layer of four neurons, one hidden layer of five neurons, and a single output unit. Constant bias terms x_0 and $z_0^{(1)}$ are added to all but the last layer, and the weight matrices $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ for both non-input layers are also represented.

2.3.2 Training

To explain how a neural network learns, we focus on the simple feed-forward fully-connected[5] neural network presented in Figure 2.1, called a **multilayered perceptron (MLP)**. An MLP works by applying a linear transformation to an input followed by an activation function[6] to generate an output. More formally, given the input feature vector $\mathbf{x}$, the linear transformation for any non-input layer l is given by:

$$\mathbf{a}^{(l)} = \mathbf{W}^{(l)}\mathbf{x} + \mathbf{b}^{(l)}, \tag{2.5}$$

where $\mathbf{b}^{(l)}$ is the layer's bias vector term,[2] and $\mathbf{W}^{(l)} \in \mathcal{W}$. Then, the layer's activation function is applied to $\mathbf{a}^{(l)}$ and determines which of the layer's neurons should be **activated**, that is, which neuron's input is important to the process of prediction. This yields the layer's output, denoted $\mathbf{z}^{(l)}$, which can either be the final output, or the input to a subsequent layer. Compounded together, across all layers, they form the network's hypothesis function $h_{\mathcal{W}}$.

The act of training is to optimise the learnable parameters (Section 2.2), here $\mathcal{W}$, such that some loss function is minimised. We do this to ultimately output y (e.g., the score of the essay that has been input) during testing. This is also true for neural networks, although

[5] For precise definitions of connectivity, refer to Diestel (2017, Section 1.4).

[6] Activation functions are typically either hard thresholds, or logistic functions. See Russell and Norvig (2003, Figure 18.17) for reference.

the way in which this is achieved differs slightly. In short, we use a learning mechanism called **back-propagation** which allows the MLP to iteratively adjust its weights during training (Russell and Norvig, 2003, Figure 18.24). This method generally uses gradient descent as its optimisation strategy, which completes our ML triptych (Section 2.2).

2.3.3 Fine-Tuning

In practice, the architecture of the neural network will depend on the nature of the task at hand, as well as that of the inputs and outputs. The number of units in every layer, the number of hidden layers, and the nature of the activation functions, can all be changed. These are some of the network's **hyper-parameters**, which also include:

(1) the learning rate α defines by how much a network updates its parameters during training as we saw in Equation (2.4);

(2) the number of **epochs** is the number of times the network will pass through the entirety of the training set during training;

(3) the **sequence length** s caps the length of the essays that will be input to the model;

(4) the **batch size** refers to the number of data entries given to the network before the network updates its weights according to a loss.

Hyper-parameters determine how the network is trained. The above is not an exhaustive list but rather some of the most common which we will use in our experiments. For an in-depth discussion, see Goodfellow et al. (2016, Chapter 8).

The process of **hyper-parameter optimisation** or **tuning**, which consists in finding the set of optimal hyper-parameters for a model, is an integral part of the development of an artificial neural network. Here, we will refer to it by the broader term of **fine-tuning** (Jurafsky and Martin, 2021, Section 11.3). For a survey of hyper-parameter optimisation algorithms, see Yu and Zhu (2020). There exists many facets to this practice, but none particularly stand out as the best within AA (Mayfield and Black, 2020, Section 3). In this study, we adopt the simplest: we will first manually initialise a model's hyper-parameters (using standard value ranges), train the model on our training set, and then evaluate it on a separate set of annotated data called the **validation set** (we discuss evaluation strategies in Section 3.4). Then, based on the evaluation results, we will update the hyper-parameter values, re-train the model and re-evaluate it on the training set. We will repeat this process in a sort of binary search of the best set of hyper-parameter values.[7] Only upon settling for one such set will we test our model (Section 2.1).

[7] Note that what we call the best model setting is potentially not actually the best in the whole world of possibilities, but one that does emerge as better than the ones we have tested.

2.3.4 Regularisation

Over-fitting occurs when a model learns a hypothesis function that is too closely fitted to the training data (Ying, 2019). We defined the act of learning as finding the best function to describe our training data but learning too well means that our model cannot generalise well when presented with new data (e.g., the test set) defeating its original intended purpose. This can happen for a number of reasons: for example, when trained for too long (i.e., the number of epochs is too high) or when the model is too complex and starts to account for the **noise** (irrelevant information) in the training set.

Regularisation methods typically address this problem by reducing the complexity of the network during training. For transformers, we can use **Dropout** (Srivastava et al., 2014). This method consists in dropping some units at random in each layer which compels the nodes to learn to fix the mistakes of other nodes. Dropout can also be applied to a single layer in **dropout layers**: these ignore some parts of their inputs during training with a probability defined by the **dropout rate**. Another way to reduce complexity is to set the **weight decay** hyper-parameter. Suppose we want to add all the parameters (weights) of our model to the loss function to penalise complexity. Since weights can be both negative and positive, we take their square, sum them together, and then multiply the sum by a small number (the weight decay) to control how large this number is compared to the loss, and finally add it to the loss. There exists many more regularisation strategies out there, but we will only refer to these two in our experiments. For a full survey on regularisation, see Moradi et al. (2020).

In this section, we gave a very brief introduction to neural networks. For a more detailed overview, we refer the reader to Goodfellow et al. (2016, Chapter 6) and Hastie et al. (2009, Chapter 11). Next, we look at how we can draw on the power of neural networks to learn features from linguistic data and finally shed some light on the nature of our model inputs: how do we go from written essays to feature vectors $\mathbf{x}$?

2.4 Word Embeddings

It is well known that computers, and by extension models, only understand and manipulate numerical representations (numbers, vectors, matrices, etc.). However, as suggested in the name, NLP demands that we process natural language (sequences of letters and symbols). We thus need some way to convert words into numbers, or more specifically, into vectors. As with traditional vectors, these could then be subjected to many kinds of mathematical operations—from simple addition and subtraction, to complex similarity measures, and many more—making them particularly easy to integrate with existing ML algorithms and techniques (Almeida and Xexéo, 2019).

Word embeddings have become central to the development of NLP (Camacho-Collados and Pilehvar, 2020). They provide intuitive, powerful and efficient feature representations of language, and are formally defined by Almeida and Xexéo (2019) as:

> **Definition 2.4.1** (Word embeddings)**.** Dense, distributed, fixed-length word vectors, built using word co-occurrence statistics as per the **distributional hypothesis**.

Here, the distributional hypothesis is the idea that the meaning of a word can be inferred from the contexts in which it appears, without necessitating any knowledge of the real world (Jurafsky and Martin, 2021). As Firth (1957) puts it: "a word is characterised by the company it keeps". So, words which appear in similar contexts may display similar meanings. This suggests that a word's embedding does not need to be specified by hand and can instead be *learned* on a training corpus (Almeida and Xexéo, 2019), paving the way for neural-based embeddings.

2.4.2 Static Embeddings

The CBOW and Skip-Gram neural network models proposed by Mikolov et al. (2013a) were some of the first neural approaches for word embeddings. They were part of the Word2Vec statistical algorithm which could learn word embeddings based on local statistics (Mikolov et al., 2013a,b). See 2.2. Alternatively, Pennington et al. (2014) proposed the GloVE algorithm which focused instead on word co-occurrences across a corpus (global statistics). However useful these approaches turned out to be, they could only compute a fixed vector representation for each given word, i.e., **static word embeddings**. For example, given "great, blue *waves*" and "she *waves* goodbye", *waves* would be represented by a single static embedding, even though it is used in two different sentences, in two different ways (noun and verb), and with two different meanings (the curling of a body of water and the act of moving one's hand to and fro).

2.4.3 Contextual Embeddings

Various attempts to generate context-dependent word representations (Neelakantan et al., 2014; Melamud et al., 2016; McCann et al., 2017) were made, culminating in the emergence of **contextual embeddings** such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019). Unlike static embeddings, every word is assigned to an individual contextual embedding based on the direct context it is used in. Coming back to our previous example, each use of *waves* would be assigned a different representation. By encoding the syntactic and semantic properties of words in context, they capture deep linguistic knowledge that can be transferred across languages (Liu et al., 2020).

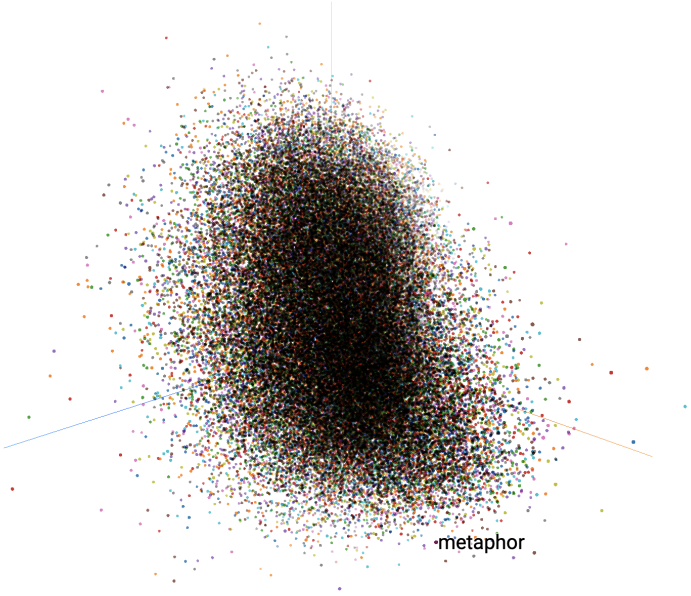

Figure 2.2: Plot of all Word2Vec (Mikolov et al., 2013a) pre-trained word embeddings (71,291 word vectors long of 200 dimensions) reduced to three-dimensions using Principal Component Analysis (PCA)[9]. Each point in space is associated to a single word, and coloured by the number of occurrences (count) of that word in the original training corpus.

Once learned, word embeddings can be used to identify and understand words encountered in new tasks. Such **pre-trained** word embeddings are now commonly used in ML models (Wang et al., 2022) and have significantly helped improve the performance of downstream NLP applications such as named-entity recognition (Pennington et al., 2014), part-of-speech tagging (Collobert et al., 2011), question answering (Xiong et al., 2017), but also automated essay assessment (Alikaniotis et al., 2016; Cozma et al., 2018). But, what if we encounter new words during testing? We cannot possibly compute embeddings for them as we have been doing, yet we still need some way to manipulate the language. This is the **out-of-vocabulary (OOV)** problem (Schuster and Nakajima, 2012). Our solution is to use **tokens**.

2.4.4 Tokenisation

Manning et al. (2008, Section 2.2.1) defines a token as "a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing". Hence, **tokenisation** is the process of splitting a text into tokens. Once we obtain a split, every unique token is assigned a unique embedding, called an **id**, to form a **vocabulary** (a sort of look-up table). By converting all tokens to their respective ids, we obtain a fully numerical representation of the original input text, which can be passed into a neural neural network for example.

[9] The image was generated using TensorBoard's Embedding Projector which can be accessed from: https://www.tensorflow.org/tensorboard/tensorboard_projector_plugin.

Traditional approaches mostly looked at tokens as words (Manning et al., 2008, Section 2.2.1), which is not enough to address the OOV problem (Schuster and Nakajima, 2012). Modern tokenisation techniques, such as **WordPiece** (Schuster and Nakajima, 2012) and **SentencePiece** (Kudo and Richardson, 2018), now split long, complex or rare words into smaller parts, sometimes throwing away certain characters (e.g., punctuation) to form sub-word tokens. These are generally assumed to solve the OOV problem (Moon and Okazaki, 2021) since smaller tokens are more likely to have been seen previously.

In this work, we will be using some version of the **Byte Pair Encoding (BPE)**. This approach looks for the most common pair of consecutive **bytes** (letters or symbols) within a text and replaces this pair with a new single unused character (byte), repeating the process until no further compression is possible. Originally introduced by Gage (1994), BPE was later adapted by Sennrich et al. (2016) to open-vocabularies, allowing one to obtain embeddings for novel words. Tokenisation is an integral part pre-processing in NLP, where obtaining a corpus which encompasses all words in all their meanings in all possible contexts is infeasible.[10]

In the next section, we briefly introduce the **transformer** architecture which is one of the principal building blocks for the model we will be presenting in this study, and behind many of the models for contextual embeddings.

2.5 Transformers

First introduced by Vaswani et al. (2017), the transformer architecture has since become the foundation for many state-of-the-art NLP models (Lin et al., 2022), including the popular BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019) models. It is based on a **self-attention mechanism**, which allows the model to focus on different parts of the input and learn the relationships between them. Thus, the transformer is able to capture the long-range dependencies needed to learn, amongst other things, contextual word embeddings.

A transformer is composed of two main parts: an **encoder** and a **decoder**, on the left and on the right of Figure 2.3 respectively, and a big advantage of transformers is these can be separated and used as two independent models. Before passing data to the encoder, the input is pre-processed and converted to a first numerical representation using tokenisation (e.g., BPE). Then, the tokenised input is passed to an attention-based encoder which generates the context-dependent representations for each word that will be used by the decoder during training. For more details on the inner workings of the transformer, refer to Vaswani et al. (2017).

[10] This is a reference to Zipf's Law and the data sparsity problem. See Piantadosi (2014) for a discussion on the topic within NLP.

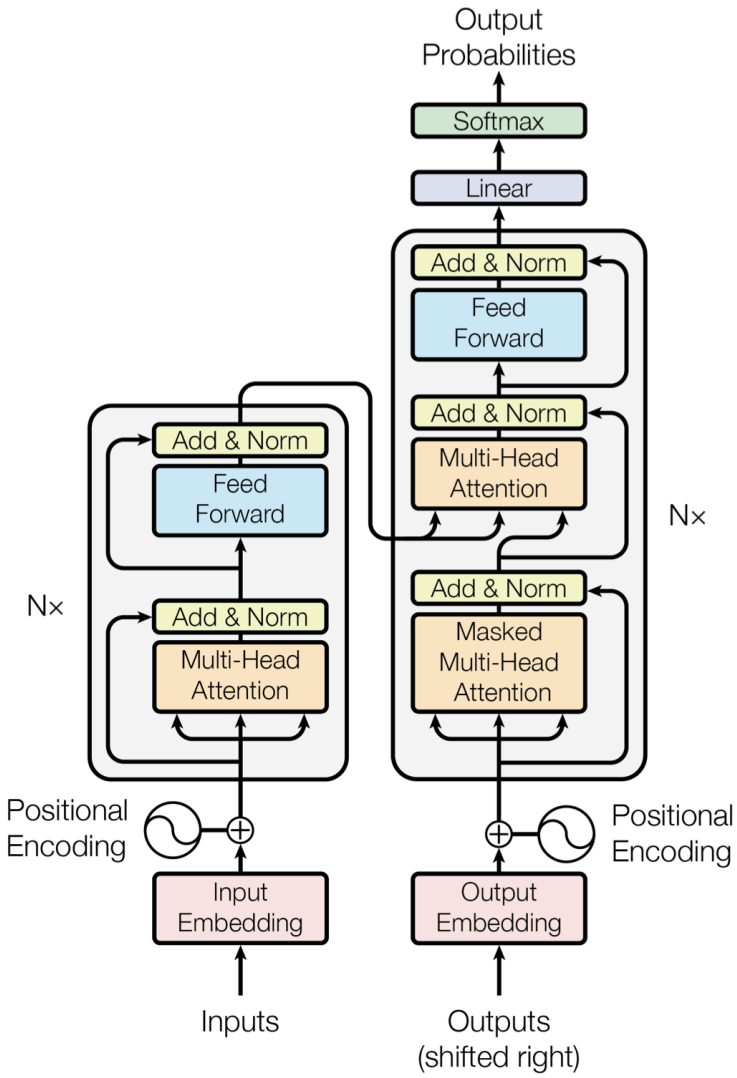

Figure 2.3: Transformer model architecture. Source: Vaswani et al. (2017).

As discussed in Section 2.4.3, relying on another algorithm to learn embedding representations for the words in our input (e.g., a training set of essays) is an instance of pre-training (Jurafsky and Martin, 2021, Section 7.4). But pre-trained word embeddings were only the beginning: when pre-trained on large enough datasets, the transformer architecture enables models to learn deep, universal language representations (Zhao et al., 2023). At this scale, the linguistic knowledge accumulated by these **large language models (LLMs)** for one type of NLP task can be reused for another (Nadas, 1984; Chen and Goodman, 1999; Thrun and Pratt, 2012; Wang et al., 2022). This is a "paradigm shift" (Sun et al., 2022). A new wave of pre-trained contextual encoders was born, upon which state-of-the-art models were rapidly developed for many downstream NLP tasks (Qiu et al., 2020; Zhou et al., 2023), including AA (Ke and Ng, 2019).

This line of work is where we begin to situate our study.

Chapter 3

Related Work

In this chapter, we position our project within the spheres of Automated Assessment (AA) and Multi-Task Learning (MTL) and capture the motivation for a system that could unite the two together.

3.1 Written Assessment

Developing deeply formative exams is hard, though many would agree that open-ended questions are generally better-suited to the task (VanderVeen et al., 2007; Graesser et al., 2009). It gives students the opportunity to actively generate knowledge, articulate difficult concepts, and engage in problem-solving (Magliano et al., 2007), which are generally not emphasised in multiple-choice exams (Magliano and Graesser, 2012). By shining a light on the strengths and weaknesses of students, instructors can provide fine-grained and personalised feedback, which is far more useful than a single overall mark for the students' growth (Shute, 2008).

Unfortunately, the cost of assessing written work weighs heavily on teachers, schools and institutions, who find themselves caught between the desire of assigning more written tasks and having to be less thorough in marking them (Miller, 2003). Further, consistency, objectivity and reliability are particularly tricky to attain for human markers, especially when marking long essays which are particularly prone to disagreements between examiners (Brown, 2010).

By threatening the frequency and quality of written assessments in education, these limitations pose a serious issue. Indeed, Stein et al. (1994, p.392) suggests that many writing disabilities are the consequence of too little time being dedicated to writing instruction and assessment. Given the importance of the writing skill in securing education and work opportunities (Council, 2013; Craighead et al., 2020), we must place written assessment at the heart of our educative systems (Defazio et al., 2010; Rao, 2019; Deane, 2022).

Automating the marking process could help us address some of these issues (Magliano and Graesser, 2012). It is what motivates this project, and the field of AA more widely. Additionally, given the importance of fine-grained feedback in a student's development (Deane, 2022; Woods et al., 2017), providing feedback is one of the drivers of this study, which has not always been addressed by the AA research community, as we will see in the next section.

3.2 Automated Assessment

The origins of AA can be traced back to the early 1960s (Daigon, 1966; Page, 1966; Page and Paulus, 1968; Page, 1994; Larkey, 1998) amidst rising issues in large-scale standardised tests such as TOEFL, IELTS and GMAT (Chodorow and Burstein, 2004; Chen et al., 2016). Issues of speed, cost, and consistency (as seen in Section 3.1), which had always been true of written assessment, began to scale (e.g., the number of candidates for the IELTS have grown exponentially over the last 30 years; Read, 2022) requiring expansive logistical efforts to mark thousands of essays under very short time-frames.

The possibility of alleviating the marking workload has made research in AA particularly attractive. As a result, an impressive body of literature on the implementation and evaluation of AA systems was developed, including Attali et al. (2008), Shermis and Burstein (2003), Burstein et al. (1998a), Burstein et al. (1998b), Burstein (2007), Burstein et al. (2010), Coniam (2009), Dickinson et al. (2012), Higgins et al. (2006), Kakkonen et al. (2004), Kakkonen and Sutinen (2008), Larkey (1998), Miller (2003), Leacock and Chodorow (2003), Phillips (2007), Williamson et al. (2012b), Phandi et al. (2015), Crossley et al. (2015), Dong and Zhang (2016), Song et al. (2020), Yang et al. (2020), Dasgupta et al. (2018), Uto et al. (2020), and Sharma et al. (2021). At first, research primarily focused on summarising the quality of an essay with a single score (e.g., the Intelligent Essay Assessor™; Landauer et al., 2003) in response to the needs of standardised tests (Phillips, 2007). As interests gained the classrooms, holistic approaches fell short in terms of providing formative feedback to students (Carlile et al., 2018).

3.2.1 Dimensions of Composition

Recent developments in Natural Language Processing (NLP) and Machine Learning (ML) have opened the door to new promising applications for research in Automated Assessment (Chen et al., 2019). The field is turning to scoring along different dimensions of quality to help students identify which aspects of their writing need improvement (Ke and Ng, 2019). Dimensions such as the detection of grammatical and mechanical errors (Kaneko et al., 2020; Chen et al., 2020; Raina et al., 2022), but also relevance to prompt (Louis and Higgins, 2010; Persing and Ng, 2014), organisation (Persing et al., 2010), coherence (Higgins et al., 2004; Miltsakaki, 2004; Burstein et al., 2010; Somasundaran et al., 2014;

Carlile et al., 2018), thesis clarity (Persing and Ng, 2013), and argument persuasiveness (Stab and Gurevych, 2014; Persing and Ng, 2015; Ke et al., 2018), have notably been studied. We include a more extensive list of the dimensions involved in the composition of an essay in Table 3.1, as we will refer to them often.

Table 3.1: Dimensions of essay quality ranked from simplest to most difficult to capture. Source: Ke and Ng (2019, Table 1).

Dimension	Description
Grammaticality	Grammar
Usage	Use of prepositions, word usage
Mechanics	Spelling, punctuation, capitalisation
Style	Word choice, sentence structure variety
Relevance	Relevance of the content to the prompt
Organisation	How well the essay is structured
Development	Development of ideas with examples
Cohesion	Appropriate use of transition phrases
Coherence	Appropriate transitions between ideas
Thesis Clarity	Clarity of the thesis
Persuasiveness	Convincingness of the major argument

In 2022, the Hewlett Foundation and the Learning Lab Agency[1] co-sponsored a competition on Kaggle called the Feedback Prize - English Language Learning,[5] in which participants were asked to design systems which could evaluate student essays along the six essay quality dimensions of their newly published ELLIPSE dataset (Section 4.2.1). Such a system could help students identify which specific dimension of their writing need improvement. Up until this point, essay dimensions had generally been studied one at a time. Motivated by recent advances in ML, our study looks at doing just what the competition demands: scoring student essays along multiple dimensions, in a hopefully novel way.

3.2.2 Machine Learning Approaches

Up until recently, the field of AA mainly focused on developing effective hand-crafted feature-based models (Craighead et al., 2020) like grammatical errors, (Yannakoudakis et al., 2011; Andersen et al., 2013), distinctive words or part-of-speech n-grams (Page and Paulus, 1968; Attali and Burstein, 2004; Bhat and Yoon, 2015; Sakaguchi et al., 2015), to predict essay scores. With the recent surge of interest in neural networks, transformer-based systems (Taghipour and Ng, 2016; Alikaniotis et al., 2016; Mayfield and Black, 2020) have gained favour. Pre-trained word embeddings (Section 2.4) now serve as input to neural networks which then perform regression to predict an essay score (Alikaniotis et al., 2016; Taghipour and Ng, 2016; Dong et al., 2017; Jin et al., 2018). Such systems

[1] See https://www.the-learning-agency-lab.com.

perform on par with previous approaches, and obviate the need for expensive feature engineering (Ke and Ng, 2019; Craighead et al., 2020; Qiu et al., 2020).

Unfortunately, neural-based approaches require large amounts of annotated training data (Zhang et al., 2021) which can be a problem for multi-dimensional essay scoring since all dimensions are not equally studied or simple (Table 3.1). We hope to address this problem using a **multi-task learning (MTL)** approach (Caruana, 1993).

3.3 Multi-Task Learning

MTL is a paradigm which "improves learning for one task by using the information contained in the training signals of other related tasks" (Caruana, 1997, Chapter 1). See Figure 3.1 for the most common example of an MTL neural network called a **hard-parameter sharing** model (Caruana, 1993; Ruder, 2017). Note that the number of hidden layers can still vary (Section 2.3), and the number of shared or task-specific layers can also be changed.[2]

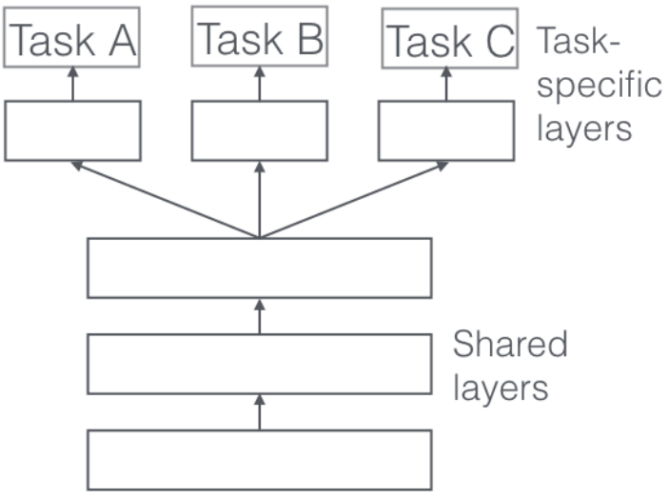

Figure 3.1: Simple multi-task learning neural network architecture. The input and hidden layers are shared, and the output layers are specific to each task. Source: Ruder (2017)

By sharing representation between similar tasks, MTL approaches can help models learn a more general, and potentially richer, set of linguistic features (Sanh et al., 2018) without relying on any real world knowledge (e.g., external linguistic annotations in NLP) at inference time (Craighead et al., 2020). This improves the **generalisation** capabilities of MTL models, that is, their capacity to adapt to previously unseen data (Caruana, 1997, Chapter 7), and boosts their performance on tasks that are generally harder, or for which we have limited amounts of data (Caruana, 1993; Andersen et al., 2021).

[2] In fact, there exists many more MTL architectures, but this is beyond the scope of this study. Refer instead to Ruder (2017) for an in-depth overview of multi-task learning neural networks.

This is particularly interesting from the perspective of multi-dimensional essay scoring which we are interested in, and suggests that MTL could help address the complexity and data imbalance between dimensions (Section 3.2.1). In fact, MTL has already been successfully applied across all applications of machine learning (Ruder, 2017) including NLP (Collobert and Weston, 2008), and more particularly to AA for grammatical error detection (Rei and Yannakoudakis, 2017), automated scoring of learner English essays Cummins and Rei (2018), as well as automated grading of transcripts of spoken language (Craighead et al., 2020). The theoretical merits of MTL, and the promise of these past results, motivate us to investigate this approach for multi-dimensional essay scoring, which, to the best of our knowledge, has not been done before.

3.4 Evaluation strategies

Within the field of AA, the evaluation of scoring systems has traditionally been carried out by comparing the systems' predicted scores to a gold standard (Section 2.1). Some of the most common metrics include:

(1) the correlation between predicted and human scores, using for instance the **Pearson** or **Spearman rank correlation coefficients** (Pearson, 1896; Spearman, 1961);

(2) error metrics such the Mean Squared Error (MSE) and the Root Mean Squared Error (RMSE) (Tyagi et al., 2022) which we encountered in Equation (2.2).

In this study, will use only the aforementioned metrics. We do this to facilitate comparison with prior research (Section 3.2). See Appendix A for the definitions of (1), and otherwise refer to Williamson et al. (2012b) and Yannakoudakis and Cummins (2015) for an extensive discussion on evaluation frameworks for AA.

In this chapter, we have situated our study within its wider research context, drawing from prior work in Education, AA and ML. We have clearly described the gap we want to address in this work, and in doing so, introduced the techniques and evaluation strategies which we will use in our upcoming experiments.

Chapter 4

Methodology

In this chapter, we describe our approach towards building a successful multi-dimensional essay scoring multi-task learning (MTL) system. We comment on our design journey in order of implementation. As such, the chapter is made up of three main sections. The first is dedicated to choosing a baseline from a series of models on a simple essay scoring task, the second to building a multi-dimensional baseline from this baseline model, and the last to establishing an MTL system atop our baseline models.

4.1 Essay Scoring Baseline

In the previous chapter (specifically Section 3.2) we presented an extensive body of Automated Assessment (AA) research upon which this study rests. Where prior work mostly focused on holistic scoring, we look at multi-dimensional essay scoring, using the very recently published ELLIPSE dataset (Crossley et al., forthcoming). This dataset has, to the best of our knowledge, not yet been used in any published work. We therefore need to construct our own original baselines for evaluation, while still anchoring ourselves in the common practices of AA. We do this in two ways: by using a classical essay scoring dataset (Ke and Ng, 2019), namely CLC FCE (Yannakoudakis et al., 2011), and employing proven machine learning (ML) methods (Section 2.5).

4.1.1 CLC FCE Dataset

The Cambridge Learner Corpus[1] (CLC), part of the Cambridge English Corpus[2] (CEC), was developed jointly by the Cambridge University Press and Cambridge Assessment. For our baseline, we used the CLC FCE dataset (Yannakoudakis et al., 2011), a collection of 1,244 exam scripts written by English language learners (ELLs) from around the world who sat the Cambridge English for Speakers of Other Languages (ESOL) First examina-

[1] To access the dictionary, see https://dictionary.cambridge.org/dictionary/learner-english/.
[2] To learn more about the initiative, refer to https://www.cambridge.org/corpus/.

tions (now known as B2 First[3]) between 2000 and 2001. The writing component of the examination consists of two essay tasks asking students to write either a letter, a report, an article, a composition or a short story, of 200 to 400 words.

The dataset includes the original written answers, transcribed and anonymised but otherwise unmodified, as well as linguistic error annotations which follow a taxonomy of 77 error types by Nicholls (2003), and some demographic data (first language, age bracket). Additionally, the corpus entries include the candidates' overall exam scores in the range 0–40 which have been fitted to a RASCH model (Fischer and Molenaar, 2012), as well as a breakdown of their individual marks for each of the two exam tasks. The authors provide little information about the latter marks. What looks like decimal grades between 0 and 5.3 are actually band scores which follow the General Impression Mark Scheme (of Cambridge. ESOL Examinations and of Cambridge. Local Examinations Syndicate, 1978, p.28). They are slightly more fine-grained than what is presented in the FCE handbook for teachers: for example, a score of 5.3 indicates the highest level within band 5, and a score of 5.1 the lowest subdivision within band 5. These slightly more fine-grained marks can be directly mapped to a 0–20 linear scale, where 0 signifies 0 and 5.3 signifies 20, which is ideal for a regression task. Similarly to Yannakoudakis et al. (2011), we use these in our experiments rather than the overall scores.

Recall that our task here is to build a model capable of learning how to automatically mark these written exam tasks (Section 2.1). Here, we will use this dataset to train, evaluate and compare a series of standard pre-trained models on a simple essay scoring regression task to establish our project's baseline. But first, we must perform the adequate steps to process and prepare the data.

4.1.2 Pre-processing

In the CLC FCE dataset, the data for each candidate is stored in individual `.xml` files. Interestingly, the linguistic error annotations are embedded into the original answers, so, in order to parse and separate the original from the corrected texts, we borrowed two helper functions[4] from Sergio (2019), and dealt with the other data fields ourselves.

For our essay score prediction task, we only need the candidates' original answers to the two exam questions and the marks that were awarded to each of them. When parsing the dataset, if the information for one of the two tasks was missing, we did not exclude the candidate altogether, and kept the single task data entry that was present. If either the original written text or the associated score was missing, the single task data entry was omitted entirely. Note that the data was not always clean (e.g., some of the task scores included additional letter characters). We stripped the marks of all such noise, and dropped those entries whose marks were still not numbers after processing.

[3] For additional information, go to https://www.cambridgeenglish.org/exams-and-tests/first/.

[4] Specifically, the functions `strip_str()` and `recursive_NS_tag_strip()` in `FCECorpusHandler.py`.

Table 4.1: Train, validation and test data split sizes (in number of task data entries) after parsing and processing the CLC FCE dataset (Section 4.1.2).

Split	Train	Validation	Test
CLC FCE	1,727	371	370

Before parsing, the CLC FCE dataset contained a total of 1,244 candidate exam scripts, for a potential of 2,488 task data entries (two exam tasks per candidate). In practice, seven essays were missing, and after performing the above pre-processing steps, our final dataset numbered 2,468 entries. These were then randomly split it into train, validation and test sets using the `train_test_split()` function of the `scikit-learn`[5] Python library. Table 4.1 shows the data split sizes (namely, 70/15/15% respective proportions of the data).

4.1.3 Models

To establish a baseline, we chose a non-exhaustive list of six transformer-based pre-trained models and their associated pre-trained tokenisers imported from the HuggingFace Transformer library[6] (Wolf et al., 2020) to be trained and fine-tuned on the previously obtained train and validation sets, and finally evaluated against one another on the test set. Here our intent was to find a good and easily reproducible baseline, and we felt that being exhaustive was unfeasible and unnecessary. As such, we favoured some of the variants of the pre-trained BERT model (Devlin et al., 2019), which have recently been used for AA by Mayfield and Black (2020), Schmalz and Brutti (2021) and Beseiso (2021), are particularly easy to use, and have, on many occasions, proven their worth to the wider NLP community. See the surveys by Wang et al. (2022) and Zhao et al. (2023).

4.1.4 Implementation

Throughout this study, all of our experiments were run using the PyTorch[7] (Paszke et al., 2019) ML framework, on NVIDIA P100 GPUs made freely available by Kaggle. Further, a random seed value of 42 was fixed for better reproducibility. Indeed, neural networks non-deterministic, and randomness can play a major role in their results (Reimers and Gurevych, 2017). This particular value was chosen in accordance with standard ML practices[8] without optimising for the best performance scores.

Each pre-trained model, and corresponding pre-trained tokeniser, was first imported from the HuggingFace Transformer library[6] setting the problem type to regression (Section 2.2). Then, using the model-specific tokeniser, we tokenised the candidate written answers

[5] For the documentation, see https://pypi.org/project/scikit-learn/.
[6] See https://huggingface.co.
[7] The library can be access from https://pypi.org/project/torch/.
[8] The number is a pop-culture reference to the popular science-fiction novel "The Hitchhiker's Guide to the Galaxy" by Adams (1995).

using each model's associated pre-trained tokeniser to obtain word embeddings which the models could manipulate. The lengths of the original essays ranged from 243 to 2,532 characters. However, due to varying model limitations,[9] we had to set a sequence length for these (Section 2.3.3). Some answers had to be truncated while others were padded to that value. Figure 4.1 shows a visual representation of this process.[10]

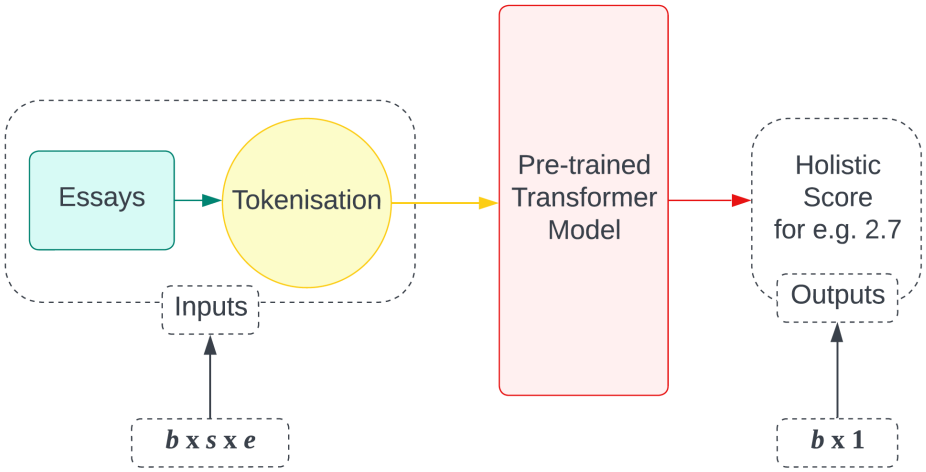

Figure 4.1: The template architecture and data flow for our holistic scoring baseline models. Here, b denotes the batch-size, s stands for the sequence length to which inputs are padded or truncated, and e is the model respective hidden embedding size.

Only after producing same-sized, model-specific embeddings for every dataset entry were we able to train the models individually on the training set using the Trainer[11] interface. By default, Trainer implements the **AdamW stochastic gradient descent** optimisation method, an **Adam algorithm** (Kingma and Ba, 2017) with weight decay fix, as introduced by Loshchilov and Hutter (2019). Details of this algorithm will not be given in this study; we only note that using models trained using AdamW optimisation has become the standard, and generally yield better results than those trained without (Loshchilov and Hutter, 2019). Further, we used each model's default regression training loss, which was generally the Mean Square Error (MSE) introduced in Equation (2.2), implemented with the `MSELoss()` function (Section 2.2) from the PyTorch library[7] (Paszke et al., 2019).

Finally, we set up Trainer such that the model weights would be saved after each epoch. At the end of training, we load the set of model parameters for which the model predicted scores are most correlated to the gold standard ones on the validation set using the Pearson coefficient. We use a correlation metric here because we follow the authors of the dataset, Yannakoudakis et al. (2011), who use both the Spearman and Pearson correlation coefficients in the evaluation of their models.[12]

[9] For example, BERT has a maximum context of 512 characters (Devlin et al., 2019).

[10] Figures 4.1, 4.5 and 4.6 were made using the free online diagramming application called LucidChart, which can be accessed from https://www.lucidchart.com/pages/.

[11] See https://huggingface.co/docs/transformers/main_classes/trainer for a full documentation.

[12] Note here that we could have equally used the Spearman Rank coefficient for the best model weights' choice.

Our next step was to identify the different models' best hyper-parameter settings by individually evaluating each of them on the validation set. As mentioned in Section 2.3, there exists a multitude of hyper-parameters. We consistently set the weight decay hyper-parameter to the standard value of 0.01[13] to keep fine-tuning as simple as possible, and only varied the learning rate in the range 0.1 to 7.0e-5, the number of epochs from 3 to 10, the sequence length in the range 32 to 512 and the batch size between 8 and 32.

Table 4.2: Final hyper-parameter values for each of the different pre-trained models we considered on the CLC FCE dataset.

Model	Epochs	LR	Batch size	Sequence length
`xlm-roberta-base`	3	5.5e-5	10	232
`distilbert-base-cased`	3	5.0e-5	16	512
`distilbert-base-uncased`	6	4.0e-5	16	300
`bert-base-uncased`	6	5.5e-5	16	232
`bert-base-cased`	6	4.0e-5	14	512
`roberta-base`	5	4.8e-5	16	500

We then picked the combination of hyper-parameters which yielded the best results[14] in terms of two different evaluation metrics: the Pearson and Spearman rank correlations (Section 3.4) between the human-marked task scores and our models' predicted ones, much as was done by the dataset's original authors Yannakoudakis et al. (2011). We also include the Root Mean Squared Error (RMSE) metric (Sections 2.2 & 3.4), a standard accuracy metric in ML (Karunasingha, 2022), for comparison. See Table 4.2 for a summary of the best hyper-parameter values we used for each model. Finally, we ran the now fine-tuned models on the test set. See Table 4.3 for the results.

Table 4.3: Performance of different fine-tuned pre-trained models on the CLC FCE test set for three metrics (rounded to 3 figures after the decimal point). The models are ranked from lowest to highest on the Pearson and Spearman's rank correlation coefficients.

Model	RMSE	Pearson	Spearman
`xlm-roberta-base`	3.059	0.516	0.475
`distilbert-base-cased`	2.495	0.618	0.588
`distilbert-base-uncased`	2.374	0.662	0.628
`bert-base-uncased`	2.284	0.670	0.629
`bert-base-cased`	**2.253**	0.672	0.651
`roberta-base`	2.354	**0.673**	**0.668**

The best results on the test set were consistently achieved using the pre-trained RoBERTa model (Liu et al., 2019). The results reported in Yannakoudakis et al. (2011, Table 1)

[13] Values of the regularisation hyper-parameter are generally kept between 0 and 0.1 (Kuhn and Johnson, 2013, p.144).

[14] For large batch sizes and/or sequence lengths, we sometimes exceeded CUDA's memory limitations, and our best results lie within those restrictions.

on the same dataset are not directly comparable because they were computed on the entire dataset. However, our correlation scores fall somewhere in their range of results. Since the object of this study is to investigate the merits of the MTL approach, and not necessarily building a state-of-the-art model, we deem our `roberta-base` regression model good enough to build our baseline from.

4.2 Multi-Dimensional Baseline

In this section, we present how we extended our simple essay scoring baseline to a multi-dimensional one; one that we could use in the evaluation of a future multi-task learning model for multi-dimensional essay scoring. But first, we present the ELLIPSE dataset which will be the basis for all of the remaining experiments.

4.2.1 ELLIPSE Dataset

The English Language Learner Insight, Proficiency and Skills Evaluation (ELLIPSE) Corpus was released by the Vanderbilt University and the Learning Agency Lab[1] in 2022 for the "Feedback Prize - English Language Learning" Kaggle competition[5] (Crossley et al., forthcoming). The public dataset contains 3,911 essays written by ELLs between the 8$^{\text{th}}$ and 12$^{\text{th}}$ grade as part of state-wide standardised writing assessments in the 2018/19 and 2019/20 school years in the United States (US). Note that the dataset includes an additional test set comprising roughly 2,700 essays used in the evaluation of the competition entries. We ignored this part of the data since it was not released to the public.

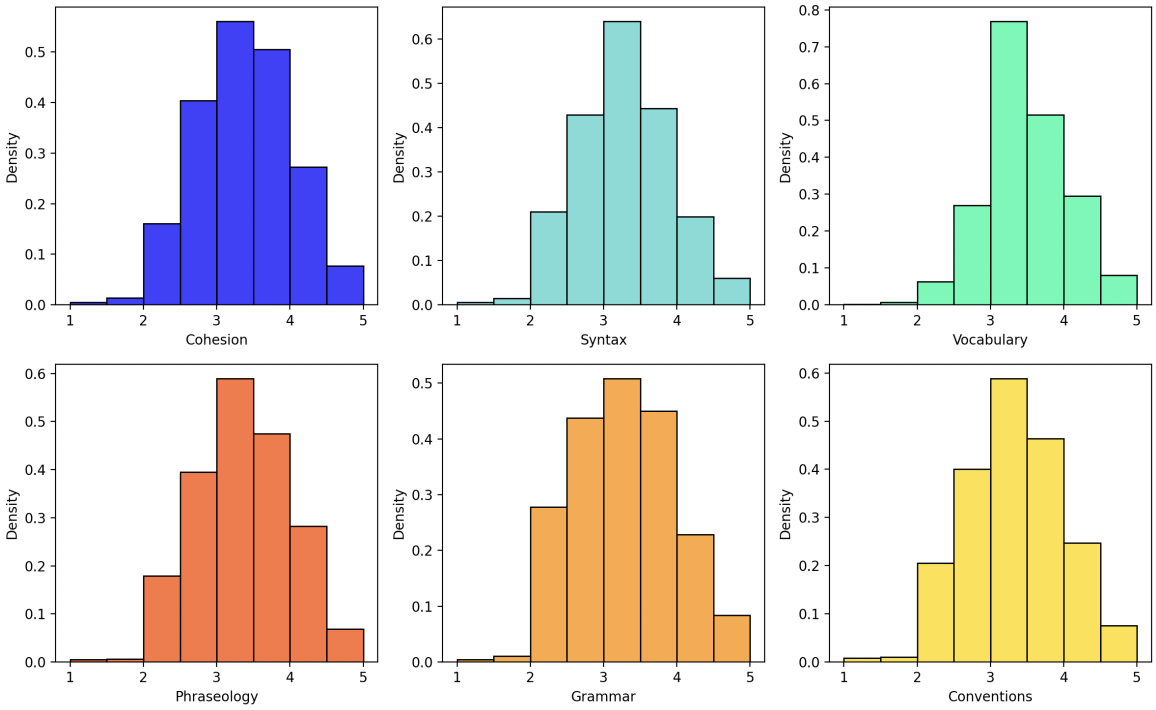

Figure 4.2: Score distribution of the different dimensions of the ELLIPSE dataset.

The specificity of this dataset is that it is multi-dimensional. All essays were independently marked by two examiners along six different dimensions of language: Cohesion, Syntax, Vocabulary, Phraseology, Grammar, and Conventions. These six dimensions were identified by teaching and research advisory boards of experts in the fields of composition and ELL education as the principal components of language acquisition (Lab, 2023).

Each essay of the ELLIPSE corpus was scored by a minimum of two trained raters. These were recruited in the Applied Linguistics and English departments of a large research university in the US, and received further training for this particular annotation task. Any disagreement between raters, defined as a difference equal to or greater than two points in a particular dimension, was adjudicated in a discussion between the two parties. The scores follow a 9-point Likert scale and range from 1.0 to 5.0 with increments of 0.5, where a maximal score in one of these dimensions signifies a native-like proficiency for that aspect of the English language. We include the dataset's annotation guidelines in Appendix B. See Figure 4.2 for the resulting score distribution of the corpus.

In the next section we look at the pre-processing steps we had to take before properly using the dataset.

4.2.2 Pre-processing

The ELLIPSE public dataset is stored as one large `.csv` file, where each line corresponds to one data entry composed of the `full_text` of a single essay identified by a unique `text_id`, along with a score for each of the six analytic measures. We parsed this file using the Python function `read_csv()` from the `pandas`[15] library (McKinney et al., 2010).

Once parsed, we inspected the data (checking for `NaN` values, incorrect data types, etc.) to find that it was clean. Thus, no entries were removed at this point.

4.2.3 Data Inspection

To the best of our knowledge, no academic paper has yet been published using ELLIPSE. Hence, our first task was to properly inspect it, and found that its essays are on average approximately 2,335 characters or 430 words long, for a maximal length of 6,044 characters or 1,260 words, and a minimal length of 239 characters or 48 words. Additionally, we computed detailed statistics for each dimension of the original dataset which we include in Table 4.4. As you can see in Figure 4.2, all of the dimensions are roughly normally distributed (Reid, 2013, p.38), and share similar parameters ($\mu \approx 3, \sigma \approx 0.65$).

Beyond length, we were interested in what the distributions of the different dimensions could reveal to us. Indeed, identifying the dimensions most closely related can potentially inform some of our future design decisions; this is particularly true in MTL where task

[15] For the documentation, see https://pypi.org/project/pandas/.

Table 4.4: Mean and standard deviation (rounded to 3 figures after the decimal point) of the different dimension distributions of ELLIPSE before (μ, σ) and after removing outliers (μ', σ') in Section 4.2.4.

Dimension	μ	σ	μ'	σ'
Cohesion	3.127	0.663	3.120	0.605
Syntax	3.028	0.644	3.015	0.577
Vocabulary	3.236	0.583	3.226	0.463
Phraseology	3.117	0.656	3.106	0.589
Grammar	3.033	0.700	3.023	0.645
Conventions	3.081	0.671	3.071	0.613

relatedness is an important topic of research (Ruder, 2017, Section 7.1). We do this using correlation metrics (Appendix A): see Figures 4.3 and 4.4 for the Pearson and Spearman rank correlation scores between the different dimensions of the ELLIPSE corpus.

	Cohesion	Syntax	Vocabulary	Phraseology	Grammar	Conventions
Cohesion	1.000000	0.622782	0.573176	0.617389	0.565320	0.590485
Syntax	0.622782	1.000000	0.572227	0.652727	0.641116	0.626624
Vocabulary	0.573176	0.572227	1.000000	0.647343	0.558494	0.568505
Phraseology	0.617389	0.652727	0.647343	1.000000	0.656495	0.588728
Grammar	0.565320	0.641116	0.558494	0.656495	1.000000	0.602700
Conventions	0.590485	0.626624	0.568505	0.588728	0.602700	1.000000

Figure 4.3: Pearson correlations between the different dimensions of the ELLIPSE dataset (rounded to 6 significant figures) using a graded colour scale.

	Cohesion	Syntax	Vocabulary	Phraseology	Grammar	Conventions
Cohesion	1.000000	0.619267	0.573901	0.614352	0.563557	0.584188
Syntax	0.619267	1.000000	0.575858	0.653159	0.639609	0.619496
Vocabulary	0.573901	0.575858	1.000000	0.654490	0.563951	0.568212
Phraseology	0.614352	0.653159	0.654490	1.000000	0.657231	0.583029
Grammar	0.563557	0.639609	0.563951	0.657231	1.000000	0.591689
Conventions	0.584188	0.619496	0.568212	0.583029	0.591689	1.000000

Figure 4.4: Spearman rank correlations (rounded to 6 significant figures).

Both the Pearson and Spearman rank correlation scores range from 0.55 to 0.66, a bracket which sits in the strong positive relationship range (Zou et al., 2003). So, the different dimensions are highly correlated making them good candidates for MTL. We note that the strongest relationships are between Phraseology, and Vocabulary and Phraseology dimensions, Grammar and Phraseology, and Syntax and Phraseology. Looking at the scoring guidelines (Appendix B), Phraseology relates to the use of proper and diverse linguistic constructions and phrases which can be linked to both the Usage and Style dimensions identified by Ke and Ng (2019) in Table 3.1. Defined so, Phraseology seems inherently linked to the lower levels and mechanics of essay composition which explains that it should be most closely related to Grammar, Syntax and Vocabulary, as opposed to, for example, Cohesion which looks at the overall organisation of an essay. This insight into the dataset will be most useful in Section 5.2.

4.2.4 Outliers

Finally, adhering to good scientific practices (Osborne and Overbay, 2004), we looked at identifying potential outliers within the dataset. Since the dimension-specific scores seem to follow a Gaussian distribution, we used the standard interquartile range (IQR) method (Chandola et al., 2009, Section 7): defining limits on the sample values that are a factor of 1.5 of the IQR below the 25^{th} percentile or above the 75^{th} percentile. Table 4.5 shows the accepted value ranges for each dimension, outside of which 296 dataset entries stand.

Table 4.5: Interquartile value ranges for the different ELLIPSE dimensions (rounded to 2 figures after the decimal point).

	Cohesion	Syntax	Vocabulary	Phraseology	Grammar	Conventions
Minimum	1.00	1.00	2.25	1.00	1.00	1.00
Maximum	5.00	5.00	4.35	5.00	5.00	5.00

Interestingly, the Vocabulary dimension was the only one to harbour outliers, mainly because of its comparatively low standard deviation (Table 4.4). There is always some debate over whether to remove outliers or not (Osborne and Overbay, 2004). On one hand, it can be beneficial to the performance of resulting models (Acuna and Rodriguez, 2004), but on the other hand, it assumes that the dimension scores each follow a normal distribution. This is not necessarily the case and indeed, these outliers seem to legitimately belong to the data, and are not intentional or unintentional errors. We decided to omit them from our working data for the main part of our experiments and later explore their impact in Section 5.3.

The remaining 3,615 entries were randomly divided into three parts: a training, validation, and testing set (in the same fashion as in Section 4.1.2). See Table 4.6 for the resulting split sizes. We also include in Table 4.4 the dimension-specific statistics of the final processed ELLIPSE dataset (after removing the outliers).

Table 4.6: Train, validation and test data split sizes (in number data entries) after parsing the ELLIPSE dataset and removing all outliers as described in Sections 4.2.2 and 4.2.4.

Split	Train	Validation	Test
ELLIPSE (no outliers)	2,530	543	542

4.2.5 Implementation

Using the essay scoring baseline we established in Section 4.1.4, we begin to build our multi-dimensional baseline. We do so by first training our `roberta-base` regression model on each of the individual essay quality dimensions of ELLIPSE, instead of the CLC FCE holistic scores. That is, for each dimension, say Cohesion, we simply adapt the previous model to predicting Cohesion scores given an input essay. We use the same tokenisation methods and the same Trainer settings. Note that by default, RoBERTa (Liu et al., 2019) uses the standard MSE loss function for training in regression.[16] See Figure 4.6 for the resulting architecture,[10] where 768 is the default hidden embedding dimension size e for the pre-trained RoBERTa model. Thus, our multi-dimensional baseline is in fact six different regression models, which were trained in much the same way as before.

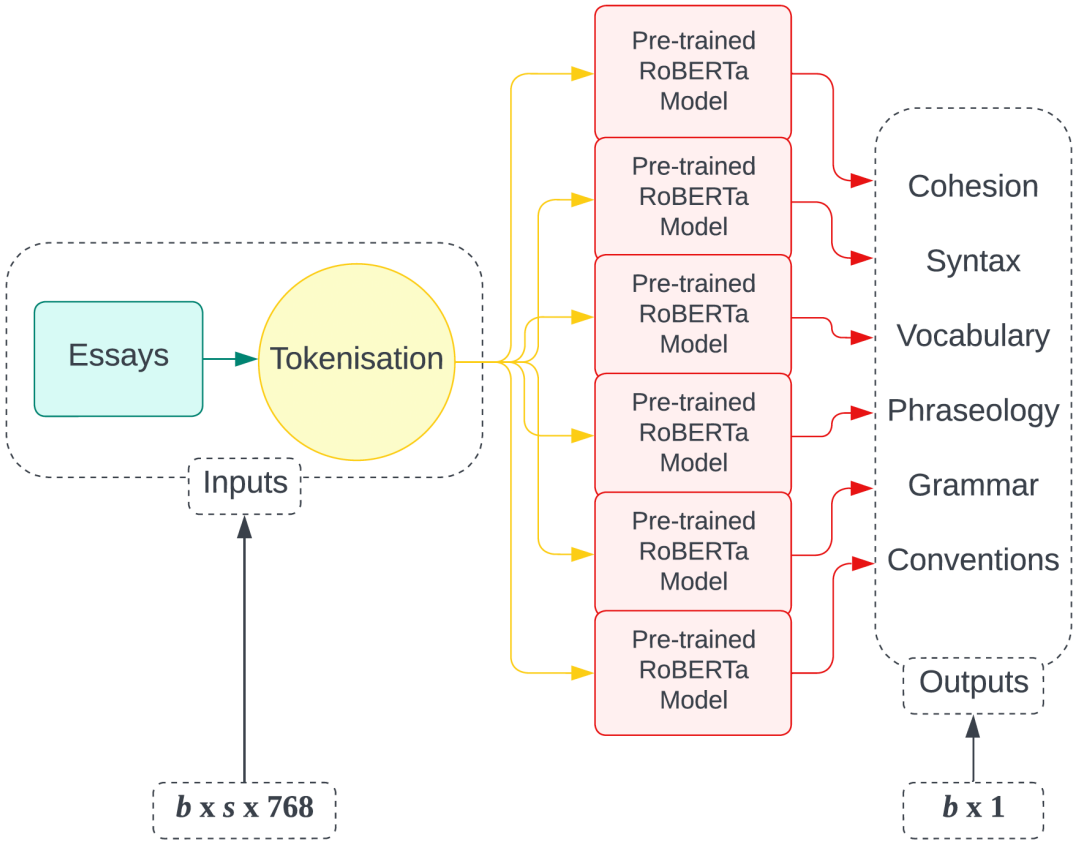

Figure 4.5: Our multi-dimensional baseline architecture and data flow. Here, b denotes the batch-size and s stands for the sequence length to which inputs are padded or truncated.

[16] See relevant documentation: https://huggingface.co/transformers/v2.9.1/model_doc/roberta.html.

Once trained, each model was individually fine-tuned on the validation set as before, using the same hyper-parameter value ranges. We ultimately picked the best settings for each dimension according to our evaluation metrics (Table 4.7). For consistency with the previous experiment, we will continue to use the Spearman and Pearson rank correlation coefficients, and the RMSE metric for evaluation. See Table 4.8 for the fine-tuned models' results on the ELLIPSE test set.

Table 4.7: Final hyper-parameter values for each of the RoBERTa regression models on the different dimensions of the ELLIPSE dataset.

Dimension	Epochs	LR	Batch size	Sequence length
Cohesion	3	5.0e-5	18	512
Vocabulary	3	6.0e-5	16	512
Phraseology	3	5.0e-5	16	375
Syntax	3	2.6e-5	16	300
Grammar	4	3.0e-5	18	280
Conventions	3	5.0e-5	20	502

Table 4.8: Performance metrics for the fine-tuned models of the multi-dimensional baseline on the ELLIPSE test set (rounded to 3 significant figures).

Dimension	RMSE	Pearson	Spearman
Cohesion	0.562	0.584	0.575
Vocabulary	**0.467**	0.599	0.605
Phraseology	0.549	0.612	0.608
Syntax	0.518	0.637	0.639
Grammar	0.490	0.676	0.676
Conventions	0.499	**0.688**	**0.681**

Notice that the best results are achieved for Conventions and Grammar. Looking at the ELLIPSE scoring rubric (Appendix B), we can equate Conventions to the Mechanics and Grammar to the Grammaticality dimensions of essay composition as established by Ke and Ng (2019) in Table 3.1, which are amongst the simplest to capture. In comparison, the Cohesion model struggles the most, and indeed, this dimension is notoriously complex (Morris and Hirst, 1991; Burstein et al., 2010; Yannakoudakis and Briscoe, 2012; Ke and Ng, 2019). We also observe that Vocabulary is awarded the smallest RMSE score, and in fact, this will be true throughout our experiments. This is likely due to the distribution of the dimension which has a much lower spread ($\sigma = 0.463$), with a shorter max/min value range than that of the other dimensions after removing the outliers values (Section 4.2.4). While RMSE is a good evaluation metric, it can only be used to compare different models or model configurations for the same variable and not between variables (as here) since it is scale-dependent (Christie and Neill, 2022, Section 8.09.2.3.2).

The six fine-tuned RoBERTa regression models form our multi-dimensional baseline. We can now move to the core part of the study: building an MTL system on the very same

dataset, in the hope of improving on this baseline. This is a rather strong baseline since all models were individually fine-tuned. Using the same overall best set of hyper-parameters across dimensions could have yielded an acceptable, although weaker baseline. However, in this study, we will use the results in Table 4.8 to evaluate the model we build next.

4.3 Multi-Task Learning Approach

Here, we describe how we built our MTL model for six-dimensional essay scoring. We use the standard hard-parameter sharing approach as presented in Figure 3.1 with one single shared encoder, and six task specific heads (one for each of the ELLIPSE dimensions).

We set the first part of our MTL model to be a `roberta-base` pre-trained encoder. This ensures the relevance of any performance comparisons we will make between our model and the multi-dimensional baseline we established in Section 4.2. The encoder's output is then passed to six identical task heads. Each task head is formed of a dropout layer for regularisation, with a dropout rate of 0.1 (Section 2.3), and a linear map of size $(e \times 1)$ which outputs the final real-valued dimension score (where recall that e is the hidden embedding dimension of RoBERTa which is equal to 768). Although the heads share the same architecture, they will be trained individually, and learn their own set of weights and parameters. See Figure 4.6 for the resulting architecture.[10]

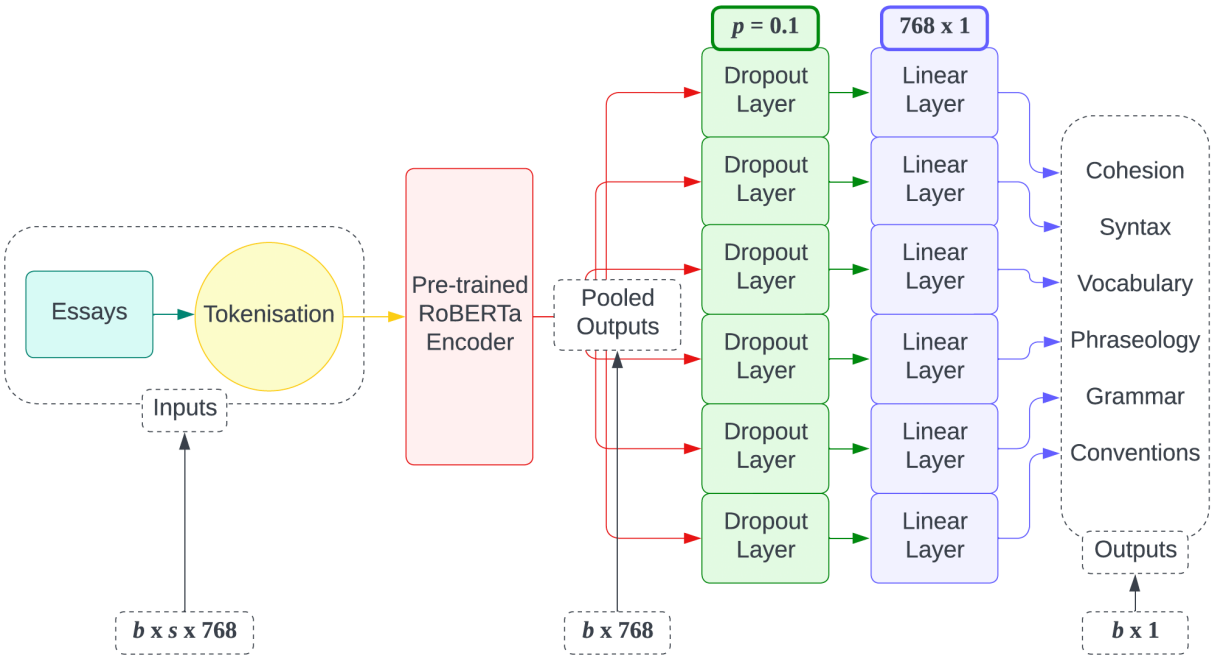

Figure 4.6: Our default multi-task learning model architecture and data flow. Here, b denotes the batch-size, s stands for the sequence length to which inputs are padded or truncated, and p is the dropout rate of the dropout layer.

Notice the type of the encoder outputs which are passed to the task-specific heads. It turns out that the pre-trained RoBERTa encoder returns two types of outputs given the tokenised inputs: **sequence** and **pooled outputs**. The first is simply an array representation of the last hidden encoder layer for each token in each sequence of the batch which will be of size $(b \times s \times e)$, whereas pooled outputs pass through an additional linear layer and a hyperbolic tangent (tanh) activation function[6] and have a $(b \times e)$ shape instead. It is generally recommended to use pooled outputs when we do not need the representations for the individual tokens because they contain contextualised information of the whole input sequence.

Now that we have our model, let us train it. We tokenise the training set essays and train our model in much the same way as in Section 4.2.5. However, because an MTL model juggles multiple tasks simultaneously, the training loss function is computed as the average of the different task head losses.[17] As previously, we use the standard MSE function as the individual task head cost functions. In our implementation, we first compute the mean of the losses for a single batch in a single task, and then after repeating the process for all six tasks, take the mean to obtain the overall loss for the batch across the six dimensions. This is the loss we wish to minimise after each batch and this is the step at which the model parameters get updated. This process is repeated again for each batch in the training set and for all training iterations (epochs).

Once our model was trained, we looked at different hyper-parameter settings in the hope of obtaining some improvements on our baseline in some, if not all, of the dimensions. We varied the learning rate, the number of epochs, the sequence length and the batch size, using the same value ranges as in Section 4.1.4. We identified four different settings of interest which we will detail in the next chapter. These are: (1) Syntax, (2) Grammar, (3) Phraseology, and lastly, (4) Conventions, Vocabulary and Cohesion. The hyper-parameter values which correspond to each one of these can be found in Table 4.9.

Table 4.9: Different hyper-parameter values for each setting identified during the fine-tuning of our MTL model.

Setting	Epochs	LR	Batch size	Sequence length
1	3	2.6e-5	16	500
2	3	2.6e-5	16	502
3	3	2.0e-5	16	500
4	4	2.7e-5	16	500

In this chapter, we described our experimental methods: from building a reliable and reproducible baseline to a MTL model. Next, we present and evaluate the results of the latter against the multi-dimensional baseline we established in Section 4.2.

[17] We use an unweighted average here, that is, all six dimensions are considered equally important in the loss.

Chapter 5

Evaluation

This chapter is dedicated to the evaluation of our multi-task learning (MTL) approach. We start by presenting and evaluating the results of our newly built MTL model on the ELLIPSE test set. Then, through some further experiments, we begin to explore and contextualise the benefits and shortcomings of our approach. Finally, we include the limitations of our experiments.

5.1 Multi-Dimensional Results

Multi-task learning is inherently a multi-objective problem (Sener and Koltun, 2019). As such, it will be difficult for us to choose one over-arching best model, as we expect to have to deal with some conflicts and trade-offs between the different dimensions. For example, during the fine-tuning of our MTL model, but also of our prior multi-dimensional baseline (Section 4.2), some dimensions seem to prefer long input sequences, others favour lower learning rates, etc. It is hard to satisfy the demands of each task. Here, we present the results of four different hyper-parameter settings (Table 4.9) which cover the best scores achieved by our MTL system across all six ELLIPSE dimensions.

As part of our evaluation, we continue to use the Pearson and Spearman's rank correlation coefficients. As mentioned in Section 3.4, these are two standard metrics in Automated Assessment (AA) as seen in, for example, Briscoe et al. (2011) and Yannakoudakis et al. (2011). We will also include the Root Mean Squared Error (RMSE) metric for reference, but will not make our evaluations based on this metric. This is mainly because it is scale-dependent (Christie and Neill, 2022, Section 8.09.2.3.2) and we are comparing dimensions which do not necessarily follow the exact same distributions (notably Vocabulary) as mentioned in Section 4.2.5. We are now ready to evaluate the results of our MTL system against the baseline we established in Section 4.2.

Table 5.1: Performance metrics for the MTL multi-dimensional scoring model on the ELLIPSE test set (rounded to 3 significant figures) using the hyper-parameter values of Setting 1 (Table 4.9). The dimensions are ranked from lowest to highest based on their average correlation scores. The lower part of the table recalls our baseline (Table 4.8) on the dimensions which are most relevant to our analysis. In green we denote the MTL results which beat the baseline, and in red the results of our MTL which are still outperformed by the baseline on the considered dimensions. We also include the differences between the MTL results, and the corresponding baseline scores.

Model	Dimension	RMSE	Pearson	Spearman
MTL	Cohesion	0.539	0.534	0.531
	Vocabulary	0.423	0.559	0.570
	Phraseology	**0.491** −0.058	**0.616** +0.004	**0.619** +0.011
	Syntax	**0.457** −0.061	**0.639** +0.002	**0.646** +0.007
	Conventions	0.495	0.671	0.665
	Grammar	0.486	0.669	0.673
Baseline	Phraseology	0.549	0.612	0.608
	Syntax	0.518	0.637	0.639

5.1.1 Setting 1: Syntax

Table 5.1 shows the results for the first setting. We notice that two of the baseline models are beaten, namely Phraseology and Syntax, although by very little. These happen to be our best recorded results for the Syntax dimension, in all of our runs on of the MTL model. On the other hand, these are not the best recorded results for Phraseology (which we will see in Setting 3).

5.1.2 Setting 2: Grammar

Table 5.2: Results on the test set for Setting 2.

Model	Dimension	RMSE	Pearson	Spearman
MTL	Cohesion	0.538	0.531	0.527
	Vocabulary	0.421	0.558	0.570
	Phraseology	**0.496** −0.053	**0.615** +0.003	**0.616** +0.008
	Syntax	0.465	0.622	0.627
	Conventions	0.498	0.661	0.658
	Grammar	**0.490** +0.000	**0.670** −0.006	**0.677** +0.001
Baseline	Phraseology	0.549	0.612	0.608
	Grammar	**0.490**	0.676	0.676

Table 5.2 presents the results for the second hyper-parameter setting, which actually include the best recorded Grammar scores for our MTL model. We see that our model surpasses the baseline on the Spearman metric for this dimension, but it does not, however, beat it for the Pearson correlation, and overall, the average correlation scores for Grammar fall under the baseline's: with 0.734 (rounded to 3 significant figures) against 0.760. As an aside, we notice again that the Phraseology baseline is outperformed.

5.1.3 Setting 3: Phraseology

Table 5.3: Results on the test set for Setting 3.

Model	Dimension	RMSE	Pearson	Spearman
MTL	Cohesion	0.545	0.530	0.531
	Vocabulary	0.432	0.555	0.572
	Syntax	0.480	0.623	0.630
	Phraseology	**0.502** −0.047	**0.625** +0.013	**0.633** +0.025
	Conventions	0.529	0.630	0.628
	Grammar	0.530	0.655	0.666
Baseline	Phraseology	0.549	0.612	0.608

We arrive to the third setting in which Phraseology achieves its very best results (see Table 5.3). We note an improvement across all three performance metrics respectively. However, looking at the other dimensions, they generally score worse than our baseline.

5.1.4 Setting 4: Conventions, Vocabulary and Cohesion

Table 5.4: Results on the test set for Setting 4.

Model	Dimension	RMSE	Pearson	Spearman
MTL	Cohesion	**0.505** −0.057	**0.551** −0.033	**0.542** −0.033
	Vocabulary	**0.377** −0.090	**0.568** −0.031	**0.576** −0.029
	Syntax	0.442	0.633	0.634
	Phraseology	0.478	0.609	0.606
	Grammar	0.465	0.668	0.671
	Conventions	**0.434** −0.065	**0.692** +0.004	**0.686** +0.005
Baseline	Cohesion	0.562	0.584	0.575
	Vocabulary	0.467	0.599	0.605
	Conventions	0.499	0.688	0.681

Finally, let us look at the results for the fourth setting in Table 5.4. This includes the best achieved results for the Conventions dimension for which we note an improvement of 0.033 over the baseline for both correlation metrics. This setting also includes the best recorded scores for Cohesion and Vocabulary, but they are far surpassed by the baseline in both correlation metrics. Interestingly however, they considerably exceed the baseline for the RMSE score, by 0.057 and 0.09 respectively.

Through the four hyper-parameter settings, we have presented the best results of the MTL model we designed to score all six ELLIPSE dimensions simultaneously. Let us now summarise our findings.

(1) First, our model is capable of surpassing our multi-dimensional baseline completely on three dimensions (Phraseology, Syntax and Conventions), and partially for all.

(2) In particular, our model beats the Phraseology baseline in three out of the four hyper-parameter settings presented. If we recall our data inspection of the dataset in Section 4.2.3, we found that the strongest correlation relationships were between the Phraseology dimension and Vocabulary, Grammar, and Syntax respectively. We suggested that this was due to the nature of Phraseology, a dimension which touches on the lower levels and mechanics of writing which plays an important part in these three dimensions specifically. The high level of relatedness of Phraseology to half of the other tasks may explain why it does so well in a MTL setting. We will reflect on this further in the next section.

(3) Finally, Vocabulary and Cohesion, which were the lowest scoring dimensions in our baseline, and are notoriously trickier than the others (Section 3.2.1), do not outperform the baseline in both correlation metrics, and seem to, instead, be doing much worse in a MTL setting. Could this be caused by one (or more) of the other dimensions?

To conclude, (1) shows definite promise for the multi-task learning approach to AA. After all, we have managed to improve on our established baseline. Unfortunately, these specific dimensions were not the most complex from the start, and we are particularly interested in improving on those that are (e.g., Cohesion). It is with this ambition in mind that we begin to explore the ideas raised in (2) and (3).

5.2 Isolating Dimensions

In the previous experiment, we have sought to create an MTL system which could predict the essay scores for all six ELLIPSE dimensions simultaneously. This approach assumes that all tasks are closely related to each other (Ruder, 2017), and indeed, taken as a whole, the six essay quality dimensions are quite similar as we saw in Section 4.2.3. This is because they all look at slightly different aspects of language in the same type of text (essays). However, taken individually, a dimension might not be as closely related to all of the other available dimensions. Take for example Cohesion and Grammar. The first refers to the overall organisation and argument structure of the essay; the second focuses on localised compliance with the rules of grammar (Appendix B). In such cases, sharing information between tasks might actually hurt performance, instead of improving it. This phenomenon is known as a **negative transfer** (Ruder, 2017).

In this section, we investigate the effects of different dimensions on one another. We will be working with smaller MTL models: instead of scoring all six dimensions simultaneously, we will focus on a subset of these using our intuition of their relatedness, the relationships we observed in Section 4.2.3, and the results of the previous section. We do not intend to be exhaustive here,[1] only to build on our previous results.

[1] In fact that would mean experimenting with 57 different models for each possible combinations of

5.2.1 Phraseology, Grammar and Syntax

Phraseology is a dimension concerned with the diversity of constructions and phrases in an essay (Appendix B), which we found to correlate highly with dimensions of Syntax and Grammar in Section 4.2.3. This was further corroborated by our results (2) in the previous section. To better understand the relationships between these dimensions, and the seemingly central role that Phraseology plays in those, we propose to compare the performance of four different MTL models, given the same hyper-parameter setting. Playing around with these three dimensions, we first isolate Grammar and Syntax individually, then each respectively with Phraseology, and finally all together. We set the hyper-parameter values to what we experimentally found to be a good consensus between the fine-tuned Grammar-Phraseology and Syntax-Phraseology MTL models to avoid using a random hyper-parameter setting. See Table 5.5.

Table 5.5: Hyper-parameter setting for our experiments in Section 5.2.1.

Epochs	LR	Batch size	Sequence length
4	2.0e-5	14	500

Table 5.6: Results of our small MTL models on the ELLIPSE test set (rounded to 3 significant figures) using the hyper-parameter setting in Table 5.5. In red we highlight the results which are below our previous MTL results for all four settings (Section 5.1, and in green those that are above.

Model	Dimension	RMSE	Pearson	Spearman
Grammar-Syntax	Grammar	**0.554**	**0.575**	**0.566**
	Syntax	**0.465**	**0.655**	**0.654**
Grammar-Phraseology	Grammar	**0.554**	**0.575**	**0.566**
	Phraseology	**0.465**	**0.655**	**0.654**
Syntax-Phraseology	Syntax	**0.527**	**0.621**	**0.614**
	Phraseology	**0.474**	**0.659**	**0.652**
All three dimensions	Grammar	**0.546**	**0.573**	**0.563**
	Syntax	**0.484**	**0.642**	**0.645**
	Phraseology	**0.424**	**0.585**	**0.599**

See Table 5.6 for the results. We see that the results for Syntax increase quite significantly when paired with the Grammar dimension, outperforming both our multi-dimensional baseline and MTL results. This remains true in the three-dimensional model, with the Phraseology dimension added. However, as is revealed by the Syntax-Phraseology results, Phraseology seems to negatively impact Syntax and in fact the results for the three-dimensional MTL model are less good than those of the Grammar-Syntax MTL model. We also note that Phraseology benefits greatly from both Syntax and Grammar, but only when individually paired with them, beating again our previous baseline and the MTL

tasks: $2^6 = 64$ is the total number of subsets in a set of six elements, from which we take the empty subset and the six singletons.

results. On the other hand, the three-dimensional setting yields worse Phraseology scores than any of our previously recorded results. Finally, it is interesting to find that Grammar helps improve the results of the other two dimensions, but never benefits from them.

We suppose that this can be explained by the nature of these three dimensions. Grammar is a fundamental aspect of language, and the features learned by our model for this dimension are likely useful for a wide range of tasks, including Phraseology and Syntax. This is not necessarily true of all the dimensions. Indeed, Phraseology and Syntax are respectively much more specific and narrow, focusing only on the diversity of phrases, and their arrangements (Appendix B). Perhaps it is precisely those dimension-specific features that are distracting from and negatively affecting Grammar. Supplemental studies are needed to determine whether these explanations are accurate and sufficient.

From these results, it becomes apparent that the relationships and interactions between different dimensions are quite complex and hard to anticipate. We also begin to see beyond the six-dimensional MTL approach where experimenting with different architectures could lead to even better results.

5.2.2 Vocabulary and Cohesion

Vocabulary and Cohesion are, apparently at least, quite different dimensions. Looking at the annotation guidelines (Appendix B), Vocabulary is strictly concerned with the proper use and diversity of words; Cohesion, on the other hand, is more complex and looks at the overall organisation of the ideas in the essay. In spite of their clear difference, they seem to react similarly to the MTL setting (Section 5.1.4). Further, looking at the hyper-parameter values for each of the fine-tuned multi-dimensional baseline models (Table 4.7), Vocabulary and Cohesion are quite similar: favouring long input sequences and low learning rates. It could be that they are negatively affected by the same dimensions, or it could be that, despite their obvious differences, they rely on similar features.

We test this by isolating the two dimensions in a single MTL model, and running the same experiment as previously. As such, we trained the model in the exact same way as in Section 4.2.5, then evaluated and hyper-parameter tuned it on the same validation set, and ultimately tested it on the same test set. The challenge of picking the best hyper-parameter setting when juggling two dimensions remains. We decided to pick the setting which yielded the best overall average correlation scores (along both Pearson and Spearman coefficients, and both dimensions). See Table 5.7 for the obtained best hyper-parameter setting, and Table 5.8 the corresponding results.

Table 5.7: Best hyper-parameter setting for our small Vocabulary-Cohesion MTL model.

Epochs	LR	Batch size	Sequence length
5	2.6e-5	16	500

Table 5.8: Results of our small Vocabulary-Cohesion MTL model on the ELLIPSE test set (rounded to 3 significant figures) using the hyper-parameter setting in Table 5.7. The lower part of the table recalls the results of the multi-dimensional baseline (Table 4.8). We use the same colouring system as in the tables of Section 5.1.

Model	Dimension	RMSE	Pearson	Spearman
Vocabulary-Cohesion MTL	Cohesion	**0.521** −0.041	**0.585** +0.001	**0.572** −0.003
	Vocabulary	**0.439** −0.031	**0.651** +0.052	**0.648** +0.043
Baseline	Cohesion	0.562	0.584	0.575
	Vocabulary	0.467	0.599	0.605

We see that new results for the Vocabulary dimension improve considerably on the baseline for all metrics, and by extension, on the previously obtained MTL model correlation scores (Table 5.4), but not the RMSE score, which achieved an all-time low in Setting 4. On the other hand, our new Cohesion results outperform the baseline on the RMSE metric, but barely edge it out on the Pearson metric, and are just surpassed in the Spearman rank. When compared to our previous MTL results (Section 5.1) however, we see a great improvement of 0.034 and 0.030 for the Pearson and Spearman correlation coefficients respectively, but again, not for the RMSE score. In fact, it is interesting to see how low the RMSE scores were for both dimensions in Setting 4 as opposed to here. We hypothesise that this could this be due to the generalisation benefits of MTL which comes with a greater diversity of tasks.

However, though our different MTL models outperform the baseline on all the dimensions, it does not do so simultaneously (for the same hyper-parameter setting). In our experiments, we have highlighted the importance of pairing the right tasks together. There are many more avenues and combinations we could have explored, but we have successfully managed to showcase some of the benefits and limitations of the MTL approach for automated essay scoring, which was after all, the intention of this study.

In the next section, we return to our decision to remove outliers from our working dataset made in Section 4.2.4.

5.3 Studying Outliers

Recall that in Section 4.2.4, we excluded 296 entries from the ELLIPSE dataset because they fell outside of the computed interquartile range (Table 4.5). We ask the following question: was removing these outliers truly beneficial to the performance of our models, and if so, at what cost? In this section, we come back on this decision and explore the impact of these outliers on the training and evaluation of our multi-dimensional baseline and MTL model.

First, we reflect on the reason why only the Vocabulary dimension contains outlier values according the IQR method. We hypothesise that it is due to the marking criteria for this specific dimension (Appendix B) with respect to the lengths of the written essays (probably due to time constraints and enforced word limits). Indeed, unlike the other dimensions, displaying extensive vocabulary range in approximately 400 words is quite challenging. Equally, it is unlikely that students will be awarded low scores (1–2) if the student writes what is deemed enough.

Inspecting the removed entries, we find that these were on average much shorter for the low-scoring ones (332 words and 2,335 characters long), and much longer for the high-scoring ones (548 words and 3,065 characters long), as opposed to the majority of essays. This is especially true for Vocabulary here, but more generally, there is a strong positive correlation between the different dimension scores and the essay lengths, a feature which has often been used in Automated Assessment (Ke and Ng, 2019, Section 3.3). From this discussion only, it is unclear whether it is legitimate to exclude these outlier values.

Let us then measure the impact of removing these outliers on the performance of our models in an experiment. We will first randomly divide the previously ignored outliers into train, validation and test sets of their own using the same split methods and proportions as previously (Sections 4.1.2 & 4.2.4). See Table 5.9 for the obtained split sizes. Then, we perform the following experiments: (1) we begin to evaluate both the multi-dimensional baseline and MTL model on the outlier test set, and finally, (2) re-train both models on a concatenation of the previous training set and the outlier training set, and evaluate them on the original test set, and the newly created outlier test set. Reporting performance in both settings will say something about the models' generalisation capabilities, having been trained with or without the outliers.

Table 5.9: Train, validation and test data split sizes (in number data entries) of the set of outliers identified in 4.2.4, and of the concatenation of the previous ELLIPSE dataset split in Table 4.6 and the outliers' split.

Split	Train	Validation	Test
Outliers	207	45	44
ELLIPSE (with outliers)	2,737	588	586

Tables 5.10 and 5.11 present the results for our multi-dimensional baseline on the original ELLIPSE test set and the newly split outliers test set respectively, using the default baseline hyper-parameters in Table 4.7. Similarly, Tables 5.12 and 5.13 show the same results for our multi-task learning model using the hyper-parameter Setting 4 (Table 4.9).

Perhaps most strikingly, both correlation metrics are consistently much higher on the outlier test set across both experiment settings and both models than anything we have previously achieved on the original test set, ranging from 0.788 to 0.955 for the baseline

Table 5.10: Multi-dimensional baseline results on the original ELLIPSE test set (rounded to 3 significant figures) using the default baseline hyper-parameter settings (Table 4.7) according to the two described experiment settings: (1) trained on the original training set (without outliers), and (2) trained on the both the original and the outlier training set. Here (1) recalls the multi-dimensional baseline in Table 4.8. We highlight in green the best achieved scores between both experiment settings, for each dimension and each metric.

	(1)			(2)		
Dimension	RMSE	Pearson	Spearman	RMSE	Pearson	Spearman
Cohesion	0.562	**0.584**	**0.575**	**0.537**	0.577	0.562
Vocabulary	**0.467**	**0.599**	**0.605**	0.631	0.595	0.600
Phraseology	0.549	0.612	0.608	**0.514**	**0.622**	**0.617**
Syntax	0.518	**0.637**	**0.639**	**0.464**	0.631	0.632
Conventions	**0.499**	**0.688**	**0.681**	0.529	0.676	0.670
Grammar	**0.490**	**0.676**	**0.676**	0.500	0.657	0.665

Table 5.11: Multi-dimensional baseline results on the outliers test set (rounded to 3 significant figures) using the default baseline hyper-parameters settings according to the two described experiment settings.

	(1)			(2)		
Dimension	RMSE	Pearson	Spearman	RMSE	Pearson	Spearman
Cohesion	**0.596**	0.898	0.841	0.651	**0.902**	**0.885**
Vocabulary	0.850	**0.955**	**0.835**	**0.796**	0.917	0.820
Phraseology	0.847	**0.902**	**0.809**	**0.663**	0.892	0.804
Syntax	0.595	**0.930**	**0.849**	**0.572**	0.928	0.847
Conventions	0.557	**0.935**	**0.910**	**0.450**	0.933	0.890
Grammar	0.591	**0.888**	0.788	**0.517**	0.883	**0.816**

Table 5.12: Multi-task learning model results on the original test set (rounded to 3 significant figures) using the Setting 4 hyper-parameter values (Table 4.9) according to the two described experiment settings. Here (1) recalls the MTL model results in Table 5.4.

	(1)			(2)		
Dimension	RMSE	Pearson	Spearman	RMSE	Pearson	Spearman
Cohesion	**0.505**	**0.551**	**0.542**	0.554	0.533	0.522
Vocabulary	**0.377**	0.568	0.576	0.491	**0.628**	**0.625**
Phraseology	**0.478**	0.609	0.606	0.502	**0.628**	**0.622**
Syntax	0.442	**0.633**	**0.634**	**0.434**	0.571	0.579
Conventions	**0.434**	**0.692**	**0.686**	0.520	0.646	0.631
Grammar	**0.465**	**0.668**	**0.671**	0.529	0.661	0.662

and from 0.778 to 0.952 for the MTL model. Inspecting the outlier test set, we computed the average standard deviation between the scores of a single entry across all dimensions,

Table 5.13: Multi-task learning model results on the outliers test set (rounded to 3 significant figures) using the Setting 4 hyper-parameter values according to the two described experiment settings.

	(1)			(2)		
Dimension	RMSE	Pearson	Spearman	RMSE	Pearson	Spearman
Cohesion	0.632	**0.922**	**0.893**	**0.497**	0.917	0.876
Vocabulary	0.685	**0.941**	**0.894**	**0.521**	0.936	0.868
Phraseology	0.735	**0.916**	0.817	**0.623**	0.897	**0.809**
Syntax	0.938	**0.952**	**0.841**	**0.723**	0.924	0.820
Conventions	0.568	**0.935**	**0.901**	**0.470**	0.929	0.856
Grammar	0.599	**0.899**	0.778	**0.506**	0.889	**0.797**

and did the same for the original test set, and found 0.32 and 0.36 (2 d.p.) respectively. Hence, the increase in performance could be related to the marks in the outlier test set being more closely distributed than in the original one. This suggests that our models generalise well to this unseen part of the data. However, we need to take into account the very small size of the outlier test set (44 entries) in comparison to the original test set (542 entries).

Now, if we compare the two experimental settings, the models trained on the original training set (without outliers) generally perform better than those trained on the full dataset (with outliers), and this is true on both the original test set, but more surprisingly on the outlier test set also. Indeed, we would have expected the models trained on the entire dataset to outperform our original models on the outlier test set but this is not overwhelmingly the case. In fact, looking at the results, the original models perform much better on both correlation metrics on the outlier test set. On the other hand, the outlier test set RMSE scores are better for the re-trained models of the experiment setting (2).

Finally, the models in (1) perform significantly better than those in (2) on the original test set in most dimensions, except for the Phraseology, as well as the Vocabulary dimension for the MTL model, and most metrics, except the RMSE scores for Syntax and Cohesion. But overall, it seems that, although we were unsure whether it was legitimate to exclude the outliers, removing them did not impede on our models' generalisation capabilities, and on the contrary, they seem to have benefited from it.

These results comfort us in our decision to remove the outliers made in Section 4.2.4, and in the reliability of the ensuing results (Sections 5.1 & 5.2), with the caveat that we did not here fine-tuned the models trained on the entire dataset in setting (2). Instead, we used the default, and best, hyper-parameter values of our baseline, and one of the best hyper-parameter settings of our MTL model, which had been found on the originally split ELLIPSE dataset (without outliers). This was important for comparability, but we might have not shown the re-trained models in their best light, and include this in our limitations (Section 5.4).

5.4 Discussion and Limitations

The finding most clearly supported by our study is the following: an MTL approach can help improve automated essay scoring in the multi-dimensional setting, and does so on a range of essay quality dimensions. Indeed, through our main experiment in Section 4.3, and further studies in Section 5.2, we have managed to improve on our multi-dimensional baseline on each of the six essay quality dimensions of ELLIPSE (namely Cohesion, Syntax, Vocabulary, Phraseology, Grammar, and Conventions), some of which are considered highly complex.

These results highlight the theoretical merits of MTL which, to the best of our knowledge, had not previously been explored in the multi-dimensional essay scoring setting of AA. At the same time, our understanding of the different essay quality dimensions, how they relate to each other, how they are similar, how they are different, is still very much limited. We will need to study them more thoroughly if we want to improve multi-dimensional essay scoring, not just in the context of MTL. Further, while we have managed to improve on all six dimensions, we have not managed to do so equally well in all of them. To fully grasp the potential of MTL, we will need properly explore how each dimension benefits from it. Given more time, we would have liked to do an in-depth qualitative analysis of our MTL models to further this agenda. Instead, we leave it for future work.

Finally, several limitations to the study can be identified.

- Since the data that is generally used as the basis for training and optimising computational models is produced by humans, using human data in evaluation is widely accepted within NLP (Kovář et al., 2016). This remains true to this day, but we note that human gold standards do present some inherent reliability problems (Williamson et al., 2012b), and have recently been put into question[2] (Basile, 2021). We must also recognise the need to evaluate AA systems using other methods, such as computing correlation scores with extrinsic metrics (e.g., state assessment scores, course grades, etc.) (Hamner and Shermis, 2012; Williamson et al., 2012a).

- Since the ELLIPSE dataset does not provide the original exam questions, we cannot do prompt-specific evaluation, which is generally standard in AA (Ke and Ng, 2019). Further, the dataset contains scripts from many different exams and sittings; the fact that we have no way to check the consistency of marking is a limitation. Indeed, it may be that the dataset marking is highly heterogeneous which is impacting its quality and the performance of our resulting models.

- Whether removing outliers was a good decision is still unclear despite our experiments in Section 5.3. We have shown that the generalisation capabilities of our models did not seem affected, but did note that fine-tuning the re-trained models could have provided further insights into the question.

[2] This discussion, however, lies well outside of this study.

Care and caution should be taken in the reading and generalisation of our results.

This chapter has presented the main findings of this study. After a series of experiments and evaluations, we have successfully showcased the benefits of the MTL approach for multi-dimensional scoring as we had set out to do. We now come to the end of our study.

Chapter 6

Conclusion

In this final chapter, we summarise the work that was presented in this report, and present some practical directions for future research in the area.

6.1 Summary

This is the first piece of work to investigate a multi-task learning (MTL) approach to multi-dimensional essay scoring within Automated Assessment (AA), and the results are promising. Indeed, we have found that MTL can help improve the overall performance of AA systems, but more importantly, that it can better the essay score predictions for a variety of essay quality dimensions, including some of the most complex (e.g., cohesion). The implications of this are considerable when we look at the existing commercial uses of automated essay scoring systems which mainly focus on holistic scoring. If we can better our multi-dimensional essay scoring systems, on all dimensions, even the complex ones, we can hope to provide a richer mark breakdown to students and teachers, which is much more suited to the classroom setting.

However, we must look beyond this. Ke and Ng (2019) argues that multi-dimensional scoring is not enough. Indeed, students who receive a low mark in a particular dimension may not know why they receive this mark, which is normally the role of feedback. Recent work by Ke et al. (2018) attempts to solve this problem by identifying the argument features which impact the persuasiveness dimension score, but overall, the area of feedback deserves more attention, and we hope that this study paves the way towards automatic generation of multi-dimensional feedback, beyond simple scores, in the future. Developing the area of feedback has highly commercially valuable applications for schools and the private sector (Ke and Ng, 2019), but more importantly, we believe that it has the potential for revolutionising the way in which people learn to write argumentative essays. In making automatic, personalised, multi-dimensional feedback available to students from all backgrounds, AA may play an increasingly important role in leveling the

existing inequities in writing instruction (Deane, 2022).

6.2 Future Work

Given more time, we would have liked to investigate the following avenues, which draw directly on some of the ideas that have been raised in this study:

(1) It is a well-known fact that essay length is a strong predictor of essay grades (Ke and Ng, 2019, Section 3.3), but did not properly investigate this idea. We would be interested in analysing the performance of our baseline and MTL models, segmenting the ELLIPSE dataset of in terms of essay length ranges, and comparing the results with length-specific systems, that is, the same models but trained and fine-tuned on subsets of the dataset (according to length ranges).

(2) Through our experiences, we felt that the MTL approach was faster and more efficient than the six models of the multi-dimensional baseline. This is simply because, in the MTL approach, we only had one model to train, evaluate, and test instead of six. Finding a way to rigorously define and measure this intuition by, for example, timing our multi-dimensional baseline versus our MTL model across all six dimensions given the same experimental settings, is an interesting avenue and the findings could further support the use of MTL in AA applications.

Further, recall that in this study, our aim was not to build the "best" multi-dimensional essay scoring system, but rather reveal the merits of the multi-task learning approach for this application. As such, there is definitely grounds to improve all the models presented here and in the hope of reaching state-of-the-art performances: by spending more time and resources in fine-tuning, but also by experimenting with a broader range of architectures. In line with this, Section 5.2 only begins to scratch the surface of the architectural possibilities of MTL, and exploring them is definitely worthy of interest.

Beyond this, a large amount of past AA effort focused on the development of features, and Ke and Ng (2019, Section 3.3) suggest that these will continue to play an important role in essay scoring systems in the future. In this study, we cast aside hand-crafted features to focus on strictly neural approaches. However, augmenting our neural models with previously identified AA features[1] to build a hybrid MTL system could produce even better results for the multi-dimensional setting.

Looking further, we identify the following possible horizons in this line of work (in no specific order of importance):

(1) Leveraging Explainable AI (XAI) techniques[2] which have been proved to work in

[1] Refer to Ke and Ng (2019, Section 3.3) for a survey of the features that have been used in AA.
[2] See Danilevsky et al. (2020) for a full survey of these techniques in NLP.

NLP, for example, LIME (Ribeiro et al., 2016), to better understand the features that a multi-task learning neural model uses to make scoring decisions, and produce explanations of its outputs, could lead to important discoveries. Additionally, it could support richer automatic feedback generation, across dimensions, which is what the field is moving towards. More generally, we hope to see more research into interpretable (Du et al., 2019) AA systems in the future.

(2) Understanding what a multi-dimensional feedback-providing application should look like and do is something that is lacking in the area of AA. There is a gap for Human-Computer Interaction (HCI) practices in this field: we would like to see user studies published, and prototypes of self-assessment and self-tutoring interfaces built, to support the future of automatic feedback generation applications. This gap has also been noted by Ke and Ng (2019, Section 9).

(3) The ELLIPSE dataset does not give access to this original exam prompts meaning that we cannot do prompt-specific evaluation. However, prompt-specific in-domain evaluation is traditional to Automated Assessment (Ke and Ng, 2019), and could reveal further insight into multi-dimensional scoring systems. Either making the original prompts available or developing a multi-dimensional dataset which includes the original prompts is a definite avenue for future work, although datasets take time. Although, collect prompt-specific data can be very costly during in dataset construction (Attali and Burstein, 2006; Yupei and Renfen, 2021).

(4) There exist many different ways of segmenting language acquisition, and we ask why these six dimension in particular? There is definitely room to explore and further segment the dimensions to provide better feedback to students, but it would be interesting to see how further granularising the essay quality dimension we are evaluating affects performance, and in particular, how it affects the MTL setting.

(5) Finally, there are many ethical considerations surrounding the topic of automated essay scoring, beyond linguistic diversity, such as fairness and bias (Madnani et al., 2017; Madnani and Cahill, 2018), that we would like to see investigated further.

Bibliography

Abbasi, I. (2020). The Influence of Technology on English Language and Literature. *English Language Teaching*, 13:1.

Acuna, E. and Rodriguez, C. (2004). On detection of outliers and their effect in supervised classification.

Adams, D. (1995). *The Hitchhiker's Guide to the Galaxy*. Pan Books.

Alikaniotis, D., Yannakoudakis, H., and Rei, M. (2016). 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. Association for Computational Linguistics.

Almeida, F. and Xexéo, G. (2019). Word embeddings: A survey.

Andersen, Ø., Yuan, Z., Watson, R., and Cheung, K. (2021). Benefits of alternative evaluation methods for automated essay scoring. In *Proceedings of the 14th International Conference on Educational Data Mining*, pages 856–864.

Andersen, Ø. E., Yannakoudakis, H., Barker, F., and Parish, T. (2013). Developing and testing a self-assessment and tutoring system. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 32–41, Atlanta, Georgia. Association for Computational Linguistics.

Arcon, N., Klein, P. D., and Dombroski, J. D. (2017). Effects of dictation, speech to text, and handwriting on the written composition of elementary school english language learners. *Reading & Writing Quarterly*, 33(6):533–548.

Arnold, K. C., Chauncey, K., and Gajos, K. Z. (2020). Predictive text encourages predictable writing. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, IUI '20, page 128–138, New York, NY, USA. Association for Computing Machinery.

Attali, Y. and Burstein, J. (2004). Automated essay scoring with e-rater® v. 2.0. *ETS Research Report Series*, 2004(2):i–21.

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

Attali, Y., Powers, D., Freedman, M., Harrison, M., and Obetz, S. (2008). Automated scoring of short-answer open-ended gre subject test items. *ETS Research Report Series*, 2008.

Basile, V. (2021). It's the end of the gold standard as we know it. In *AIxIA 2020 – Advances in Artificial Intelligence*, pages 441–453. Springer International Publishing.

Beigman Klebanov, B., Flor, M., and Gyawali, B. (2016). Topicality-based indices for essay scoring. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 63–72, San Diego, CA. Association for Computational Linguistics.

Beigman Klebanov, B., Madnani, N., and Burstein, J. (2013). Using pivot-based paraphrasing and sentiment profiles to improve a subjectivity lexicon for essay data. *Transactions of the Association for Computational Linguistics*, 1:99–110.

Beseiso, M. H. (2021). Essay scoring tool by employing roberta architecture. *International Conference on Data Science, E-learning and Information Systems 2021.*

Bhat, S. and Yoon, S.-Y. (2015). Automatic assessment of syntactic complexity for spontaneous speech scoring. *Speech Communication*, 67:42–57.

Briscoe, T., Medlock, B., and Andersen, Ø. (2011). Automated assessment of esol free text examinations.

Brown, G. (2010). The validity of examination essays in higher education: Issues and responses. *Higher Education Quarterly*, 64:276 – 291.

Burstein, J. (2007). Sentence similarity measures for essay coherence. In *Proceedings of the 7th International Workshop on Computational Semantics*, pages 77–88.

Burstein, J., Chodorow, M., and Leacock, C. (2004). Automated essay evaluation: The criterion online writing service. *AI Mag.*, 25(3):27–36.

Burstein, J., Kukich, K., Wolff, S., Lu, C., and Chodorow, M. (1998a). Computer analysis of essays. In *NCME Symposium on automated Scoring.*

Burstein, J., Kukich, K., Wolff, S., Lu, C., and Chodorow, M. (1998b). Enriching automated essay scoring using discourse marking. In *Discourse Relations and Discourse Markers.*

Burstein, J., Tetreault, J., and Andreyev, S. (2010). 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, Los Angeles, California. Association for Computational Linguistics.

Camacho-Collados, J. and Pilehvar, M. T. (2020). Embeddings in natural language processing. In *Proceedings of the 28th International Conference on Computational Linguis-*

tics: Tutorial Abstracts, pages 10–15, Barcelona, Spain (Online). International Committee for Computational Linguistics.

Carlile, W., Gurrapadi, N., Ke, Z., and Ng, V. (2018). Give me more feedback: Annotating argument persuasiveness and related attributes in student essays. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 621–631, Melbourne, Australia. Association for Computational Linguistics.

Carroll, J. (1961). The nature of the data, or how to choose a correlation coefficient. *Psychometrika*, 26(4):347–372.

Caruana, R. (1993). Multitask learning: A knowledge-based source of inductive bias. In *International Conference on Machine Learning*.

Caruana, R. (1997). Multitask learning. *Machine Learning*, 28:41–75.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3).

Chen, J., Fife, J. H., Bejar, I. I., and Rupp, A. A. (2016). Building e-rater® Scoring Models Using Machine Learning Methods. *ETS Research Report Series*, 2016(1):1–12.

Chen, M., Ge, T., Zhang, X., Wei, F., and Zhou, M. (2020). Improving the efficiency of grammatical error correction with erroneous span detection and correction.

Chen, M. X., Lee, B. N., Bansal, G., Cao, Y., Zhang, S., Lu, J., Tsay, J., Wang, Y., Dai, A. M., Chen, Z., Sohn, T., and Wu, Y. (2019). Gmail Smart Compose: Real-Time Assisted Writing. *arXiv e-prints*, page arXiv:1906.00080.

Chen, P. Y. and Popovich, P. M. (2011). Correlation: Parametric and nonparametric measures. In *Correlation*, pages 2–87. SAGE Publications, Inc., 2455 Teller Road, Thousand Oaks California 91320 United States of America.

Chen, S. F. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394.

Chen, Y.-Y., Liu, C.-L., Chang, T.-H., and Lee, C.-H. (2010). An unsupervised automated essay scoring system. *IEEE Intelligent Systems*, 25(5):61–67.

Chodorow, M. and Burstein, J. (2004). Beyond Essay Length: Evaluating e-rater®'s Performance on TOEFFL® Essays. *ETS Research Report Series*, 2004(1):i–38.

Chowdhary, P. (2020). *Natural Language Processing*, pages 603–649.

Chowdhury, G. (2005). Natural language processing. *ARIST*, 37:51–89.

Christie, D. and Neill, S. P. (2022). 8.09 - measuring and observing the ocean renewable

energy resource. In Letcher, T. M., editor, *Comprehensive Renewable Energy (Second Edition)*, pages 149–175. Elsevier, Oxford, second edition edition.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37 – 46.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA. Association for Computing Machinery.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

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

Council, B. (2013). The english effect: the impact of english, what it's worth to the UK and why it matters to the world.

Cozma, M., Butnaru, A., and Ionescu, R. T. (2018). Automated essay scoring with string kernels and word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 503–509, Melbourne, Australia. Association for Computational Linguistics.

Craighead, H., Caines, A., Buttery, P., and Yannakoudakis, H. (2020). Investigating the effect of auxiliary objectives for the automated grading of learner English speech transcriptions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2258–2269, Online. Association for Computational Linguistics.

Crossley, S., Tian, Y., Baffour, P., Franklin, A., Kim, Y., Morris, W., Benner, M., Picou, A., and Boser, U. (forthcoming). The English Language Learner Insight, Proficiency and Skills Evaluation (ELLIPSE) Corpus.

Crossley, S. A., Kyle, K., and McNamara, D. S. (2015). The tool for the automatic analysis of text cohesion (TAACO): Automatic assessment of local, global, and text cohesion. *Behavior Research Methods*, 48(4):1227–1237.

Cummins, R. and Rei, M. (2018). Neural multi-task learning in automated assessment.

Daigon, A. (1966). Computer grading of english composition. *The English Journal*, 55(1):46–52.

Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., and Sen, P. (2020). A survey of the state of explainable AI for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Lin-*

guistics and the 10th International Joint Conference on Natural Language Processing, pages 447–459, Suzhou, China. Association for Computational Linguistics.

Dasgupta, T., Naskar, A., Dey, L., and Saha, R. (2018). Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 93–102, Melbourne, Australia. Association for Computational Linguistics.

Deane, P. (2022). The importance of assessing student writing and improving writing instruction. research notes. *Educational Testing Service*.

Defazio, J., Jones, J., Tennant, F., and Hook, S. A. (2010). Academic literacy: The importance and impact of writing across the curriculum–a case study. *Journal of the Scholarship of Teaching and Learning*, 10(2):34–47.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Dickinson, M., Kübler, S., and Meyer, A. (2012). Predicting learner levels for online exercises of Hebrew. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 95–104, Montréal, Canada. Association for Computational Linguistics.

Diestel, R. (2017). *Graph Theory*. Graduate texts in mathematics. Springer, Berlin, Germany, 5 edition.

Dong, F. and Zhang, Y. (2016). 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. Association for Computational Linguistics.

Dong, F., Zhang, Y., and Yang, J. (2017). 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. Association for Computational Linguistics.

Du, M., Liu, N., and Hu, X. (2019). Techniques for interpretable machine learning.

Farra, N., Somasundaran, S., and Burstein, J. (2015). Scoring persuasive essays using opinions and their targets. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 64–74, Denver, Colorado. Association for Computational Linguistics.

Faulkner, A. (2014). Automated classification of stance in student essays: An approach using stance target information and the wikipedia link-based measure. In *The Florida AI Research Society*.

Firth, J. R. (1957). A synopsis of linguistic theory 1930-55. 1952-59:1–32.

Fischer, G. H. and Molenaar, I. W. (2012). Rasch models: Foundations, recent developments, and applications.

Gage, P. (1994). A new algorithm for data compression. *The C Users Journal archive*, 12:23–38.

Galton, F. (1877). Typical laws of heredity 1. *Nature*, 15(389):512–514.

Galton, F. (1889). I. co-relations and their measurement, chiefly from anthropometric data. *Proceedings of the Royal Society of London*, 45(273-279):135–145.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Google (2023). From Smart Reply to "Help Me Write" in Gmail Thread. `https://twitter.com/Google/status/1656344805268389911?s=20`. Published at 6:05PM on May 10th, 2023.

Graesser, A. et al. (2009). What is a good question? Threads of coherence in research on the development of reading ability.

Hall, P., Gill, N., Kurka, M., and Phan, W. (2017). Machine learning interpretability with h2o driverless ai. *H2O. ai*.

Hamner, B. and Shermis, M. D. (2012). Contrasting state-of-the-art automated scoring of essays: analysis.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer New York.

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

Higgins, D., Burstein, J., Marcu, D., and Gentile, C. (2004). Evaluating multiple aspects of coherence in student essays. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 185–192, Boston, Massachusetts, USA. Association for Computational Linguistics.

Jacob, B. (1995). Linear functions. In *Textbooks in Mathematical Sciences*, pages 1–39. Springer Berlin Heidelberg.

Jin, C., He, B., Hui, K., and Sun, L. (2018). TDNN: A two-stage deep neural network for prompt-independent automated essay scoring. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1088–1097, Melbourne, Australia. Association for Computational Linguistics.

Jupp, V. (2006). *The SAGE Dictionary of Social Research Methods*. SAGE Publications, Ltd.

Jurafsky, D. and Martin, J. (2021). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall series in artificial intelligence. Pearson Prentice Hall, third edition.

Kakkonen, T., Myller, N., and Sutinen, E. (2004). Semi-automatic evaluation features in computer-assisted essay assessment. In *Computers and Advanced Technology in Education*.

Kakkonen, T. and Sutinen, E. (2008). Evaluation criteria for automatic essay assessment systems – there is much more to it than just the correlation.

Kaneko, M., Mita, M., Kiyono, S., Suzuki, J., and Inui, K. (2020). Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction.

Karunasingha, D. S. K. (2022). Root mean square error or mean absolute error? use their ratio as well. *Information Sciences*, 585:609–629.

Ke, Z., Carlile, W., Gurrapadi, N., and Ng, V. (2018). Learning to give feedback: Modeling attributes affecting argument persuasiveness in student essays. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4130–4136. International Joint Conferences on Artificial Intelligence Organization.

Ke, Z. and Ng, V. (2019). Automated essay scoring: A survey of the state of the art. In *International Joint Conference on Artificial Intelligence*.

Kendall, M. G. (1938). A New Metric of Rank Correlation. *Biometrika*, 30(1-2):81–93.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Kovář, V., Jakubíček, M., and Horák, A. (2016). On evaluation of natural language processing tasks - is gold standard evaluation methodology a good solution? pages 540–545.

Kudo, T. and Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. Springer New York.

Lab, T. L. A. (2023). The feedback prize: A case study in assisted writing feedback tools working paper.

Landauer, T., Laham, D., and Foltz, P. (2003). Automated scoring and annotation of essays with the intelligent essay assessor. *Automated essay scoring: A cross-disciplinary perspective*, pages 87–112.

Larkey, L. S. (1998). Automatic essay grading using text categorization techniques. In

Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Leacock, C. and Chodorow, M. (2003). Crater: Automated scoring of short-answer questions. *Language Resources and Evaluation - LRE*, 37:389–405.

Li, Z., Link, S., Ma, H., Yang, H., and Hegelheimer, V. (2014). The role of automated writing evaluation holistic scores in the esl classroom. *System*, 44:66–78.

Lin, P., Van Brummelen, J., Lukin, G., Williams, R., and Breazeal, C. (2020). Zhorai: Designing a conversational agent for children to explore machine learning concepts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:13381–13388.

Lin, T., Wang, Y., Liu, X., and Qiu, X. (2022). A survey of transformers. *AI Open*, 3:111–132.

Liu, Q., Kusner, M. J., and Blunsom, P. (2020). A survey on contextual embeddings.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization.

Louis, A. and Higgins, D. (2010). Off-topic essay detection using short prompt texts. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 92–95, Los Angeles, California. Association for Computational Linguistics.

Madnani, N. and Cahill, A. (2018). Automated scoring: Beyond natural language processing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1099–1109, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Madnani, N., Loukina, A., von Davier, A., Burstein, J., and Cahill, A. (2017). Building better open-source tools to support fairness in automated scoring. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 41–52, Valencia, Spain. Association for Computational Linguistics.

Magliano, J. P. and Graesser, A. C. (2012). Computer-based assessment of student-constructed responses. *Behavior Research Methods*, 44(3):608–621.

Magliano, J. P., Millis, K., Ozuru, Y., and McNamara, D. S. (2007). A multidimensional framework to evaluate reading assessment tools. *Reading comprehension strategies: Theories, interventions, and technologies*, pages 107–136.

Manning, C. D., Raghavan, P., and Schutze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, England.

Markauskaite, L., Marrone, R., Poquet, O., Knight, S., Martinez-Maldonado, R., Howard, S., Tondeur, J., De Laat, M., Buckingham Shum, S., Gašević, D., and Siemens, G. (2022). Rethinking the entwinement between artificial intelligence and human learning: What capabilities do learners need for a world with ai? *Computers and Education: Artificial Intelligence*, 3:100056.

Mayfield, E. and Black, A. W. (2020). Should you fine-tune BERT for automated essay scoring? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 151–162, Seattle, WA, USA → Online. Association for Computational Linguistics.

McCann, B., Bradbury, J., Xiong, C., and Socher, R. (2017). Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, 30.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133.

McKinney, W. et al. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX.

Melamud, O., Goldberger, J., and Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality.

Miller, T. (2003). Essay assessment with latent semantic analysis. *Journal of Educational Computing Research*, 29(4):495–512.

Miltsakaki, E. (2004). Evaluation of text coherence for electronic essay scoring systems. *Natural Language Engineering*, 10:25 – 55.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill.

Moon, S. and Okazaki, N. (2021). Effects and mitigation of out-of-vocabulary in universal language models. *J. Inf. Process.*, 29:490–503.

Moradi, R., Berangi, R., and Minaei, B. (2020). A survey of regularization strategies for deep models. *Artificial Intelligence Review*, 53.

Morris, J. and Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Nadas, A. (1984). Estimation of probabilities in the language model of the ibm speech recognition system. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(4):859–861.

Neelakantan, A., Shankar, J., Passos, A., and McCallum, A. (2014). Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar. Association for Computational Linguistics.

Neter, J., Kutner, M. H., Nachtsheim, C. J., and Wasserman, W. (1996). *Applied Linear Statistical Models*. Irwin, Chicago.

Nguyen, H. V. and Litman, D. J. (2018). Argument mining for improving the automated scoring of persuasive essays. In *AAAI Conference on Artificial Intelligence*.

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

North, B. and Piccardo, E. (2020). *Common European Framework of Reference for Languages: Learning, Teaching, Assessment Common European Framework of Reference for Languages: Learning, Teaching, Assessment. Companion volume Language Policy Programme Education Policy Division Education Department Council of Europe*.

of Cambridge. ESOL Examinations, U. and of Cambridge. Local Examinations Syndicate, U. (1978). *First Certificate in English: Handbook for Teachers for Examinations from December 2008*. Experts in language assessment. Cambridge University Press.

Osborne, J. W. and Overbay, A. (2004). The power of outliers (and why researchers should always check for them).

Page, E. B. (1966). The imminence of... grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243.

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

Page, E. B. and Paulus, D. H. (1968). The analysis of essays by computer. final report.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library.

Pearson, K. (1896). VII. mathematical contributions to the theory of evolution.—III.

regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 187:253–318.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Persing, I., Davis, A., and Ng, V. (2010). Modeling organization in student essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 229–239, Cambridge, MA. Association for Computational Linguistics.

Persing, I. and Ng, V. (2013). 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, Sofia, Bulgaria. Association for Computational Linguistics.

Persing, I. and Ng, V. (2014). 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, Baltimore, Maryland. Association for Computational Linguistics.

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

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Phandi, P., Chai, K. M. A., and Ng, H. T. (2015). Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, Lisbon, Portugal. Association for Computational Linguistics.

Phillips, S. (2007). *Automated essay scoring*. desLibris.

Piantadosi, S. T. (2014). Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21(5):1112–1130.

Pomerleau, D. A. (1988). Alvinn: An autonomous land vehicle in a neural network. In *NIPS*.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., and Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Raina, V., Lu, Y., and Gales, M. (2022). Grammatical error correction systems for automated assessment: Are they susceptible to universal adversarial attacks? In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 158–171, Online only. Association for Computational Linguistics.

Rao, P. (2019). The Role of English as a Global Language. *Research Journal of English*, 4:65–79.

Read, J. (2022). Test review: The international english language testing system (ielts). *Language Testing*, 39(4):679–694.

Rei, M. and Yannakoudakis, H. (2017). Auxiliary objectives for neural error detection models. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 33–43, Copenhagen, Denmark. Association for Computational Linguistics.

Reid, H. M. (2013). *Introduction to statistics*. SAGE Publications, Thousand Oaks, CA.

Reimers, N. and Gurevych, I. (2017). Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks.

Rudner, L. and Liang, T. (2002). Automated essay scoring using bayes' theorem. *Journal of Technology, Learning, and Assessment*, 1.

Russell, S. and Norvig, P. (2003). Artificial intelligence: A modern approach, 2/e. *Pretence artificial Hall series in intelligence, Chapter Intelligent Agent*, pages 31–52.

Sakaguchi, K., Heilman, M., and Madnani, N. (2015). Effective feature integration for automated short answer scoring. In *Proceedings of the 2015 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1049–1054, Denver, Colorado. Association for Computational Linguistics.

Sanh, V., Wolf, T., and Ruder, S. (2018). A hierarchical multi-task approach for learning embeddings from semantic tasks.

Schmalz, V. J. and Brutti, A. (2021). Automatic assessment of english cefr levels using bert embeddings. In *Proceedings of the Eighth Italian Conference on Computational Linguistics*.

Schuster, M. and Nakajima, K. (2012). Japanese and korean voice search. pages 5149–5152.

Sener, O. and Koltun, V. (2019). Multi-task learning as multi-objective optimization.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units.

Sergio, G. C. (2019). gcunhase/FCECorpusXML: Converting FCE Corpus from XML to TXT format.

Shardlow, M., Cooper, M., and Zampieri, M. (2020). CompLex — a new corpus for lexical complexity prediction from Likert Scale data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 57–62, Marseille, France. European Language Resources Association.

Shardlow, M., Evans, R., Paetzold, G. H., and Zampieri, M. (2021). SemEval-2021 task 1: Lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.

Sharma, A., Kabra, A., and Kapoor, R. (2021). Feature enhanced capsule networks for robust automatic essay scoring. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track*, pages 365–380. Springer International Publishing.

Shermis, M. D. and Burstein, J. C. (2003). *Automated essay scoring: A cross-disciplinary perspective*. Routledge.

Shute, V. J. (2008). Focus on formative feedback. *Review of educational research*, 78(1):153–189.

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

Song, W., Zhang, K., Fu, R., Liu, L., Liu, T., and Cheng, M. (2020). Multi-stage pre-training for automated Chinese essay scoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6723–6733, Online. Association for Computational Linguistics.

Spearman, C. (1961). The proof and measurement of association between two things.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Stab, C. and Gurevych, I. (2014). Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Stein, M., Isaacson, S., and Dixon, R. C. (1994). Effective writing instruction for diverse learners. *School Psychology Review*, 23(3):392–405.

Sullivan, G. and Feinn, R. (2012). Using effect size—or why the p value is not enough. *Journal of graduate medical education*, 4:279–82.

Sun, T.-X., Liu, X.-Y., Qiu, X.-P., and Huang, X.-J. (2022). Paradigm shift in natural language processing. *Machine Intelligence Research*, 19(3):169–183.

Taghipour, K. and Ng, H. T. (2016). 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. Association for Computational Linguistics.

Thrun, S. and Pratt, L. (2012). *Learning to learn*. Springer Science & Business Media.

Turing, A. M. (1950). I.—Computing Machinery and Intelligence. *Mind*, LIX(236):433–460.

Tyagi, K., Rane, C., Harshvardhan, and Manry, M. (2022). Chapter 4 - regression analysis. In Pandey, R., Khatri, S. K., kumar Singh, N., and Verma, P., editors, *Artificial Intelligence and Machine Learning for EDGE Computing*, pages 53–63. Academic Press.

Uto, M., Xie, Y., and Ueno, M. (2020). Neural automated essay scoring incorporating handcrafted features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Vajjala, S. (2017). Automated assessment of non-native learner essays: Investigating the role of linguistic features. *International Journal of Artificial Intelligence in Education*, 28(1):79–105.

VanderVeen, A., Huff, K., Gierl, M., McNamara, D. S., Louwerse, M., and Graesser, A.

(2007). Developing and validating instructionally relevant reading competency profiles measured by the critical reading section of the sat reasoning test™.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

Wang, H., Li, J., Wu, H., Hovy, E., and Sun, Y. (2022). Pre-trained language models and their applications. *Engineering*.

Wang, Y., Wei, Z., Zhou, Y., and Huang, X. (2018). Automatic essay scoring incorporating rating schema via reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 791–797, Brussels, Belgium. Association for Computational Linguistics.

Wen, X. and Walters, S. M. (2022). The impact of technology on students' writing performances in elementary classrooms: A meta-analysis. *Computers and Education Open*, 3:100082.

Williamson, D., Xi, X., and Breyer, F. (2012a). A framework for evaluation and use of automated scoring. *Educational Measurement: Issues and Practice*, 31:2 – 13.

Williamson, D. M., Xi, X., and Breyer, F. J. (2012b). A Framework for Evaluation and Use of Automated Scoring. *Educational Measurement: Issues and Practice*, 31:2–13.

Wilson, J. and Roscoe, R. D. (2020). Automated writing evaluation and feedback: Multiple metrics of efficacy. *Journal of Educational Computing Research*, 58:125 – 87.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Woods, B., Adamson, D., Miel, S., and Mayfield, E. (2017). Formative essay feedback using predictive scoring models. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Xiong, C., Zhong, V., and Socher, R. (2017). Dcn+: Mixed objective and deep residual coattention for question answering. *arXiv preprint arXiv:1711.00106*.

Yang, R., Cao, J., Wen, Z., Wu, Y., and He, X. (2020). Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, Online. Association for Computational Linguistics.

Yannakoudakis, H. and Briscoe, T. (2012). Modeling coherence in ESOL learner texts.

In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 33–43, Montréal, Canada. Association for Computational Linguistics.

Yannakoudakis, H., Briscoe, T., and Medlock, B. (2011). A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Yannakoudakis, H. and Cummins, R. (2015). 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, Denver, Colorado. Association for Computational Linguistics.

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168(2):022022.

Yu, T. and Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications.

Yupei, W. and Renfen, H. (2021). A prompt-independent and interpretable automated essay scoring method for Chinese second language writing. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1202–1217, Huhhot, China. Chinese Information Processing Society of China.

Zhang, Y., Tino, P., Leonardis, A., and Tang, K. (2021). A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. (2023). A survey of large language models.

Zhou, C., Li, Q., Li, C., Yu, J., Liu, Y., Wang, G., Zhang, K., Ji, C., Yan, Q., He, L., Peng, H., Li, J., Wu, J., Liu, Z., Xie, P., Xiong, C., Pei, J., Yu, P. S., and Sun, L. (2023). A comprehensive survey on pretrained foundation models: A history from bert to chatgpt.

Zou, K. H., Tuncali, K., and Silverman, S. G. (2003). Correlation and simple linear regression. *Radiology*, 227(3):617–628.

Appendix A

Correlation Metrics

Throughout the study, we encountered the concept of correlation metrics that we did not formally define at the time. This chapter is designed to fill that gap.

A.1 Correlation

The word **correlation** is used in everyday life to denote some form of association. It is at the same time one of the most widely used and frequently misused statistic (Carroll, 1961; Chen and Popovich, 2011). Jupp (2006) defines it as:

> **Definition A.1.1** (Correlation)**.** A linear relationship between two numerical variables, usually denoted as $\mathbf{x}$ and $\mathbf{y}$. The value of the correlation coefficient lies between $+1$ and -1. A positive coefficient indicates that a high value of $\mathbf{x}$ tends to be associated with a high value of $\mathbf{y}$ and a negative coefficient indicates that as the value of $\mathbf{x}$ increases the value of $\mathbf{y}$ is likely to decrease. A coefficient of 0 means that there is no relationship between the two variables.

Following previous Automated Assessment literature (Briscoe et al., 2011; Yannakoudakis et al., 2011), we will only use the Pearson and Spearman's rank coefficients, but there exists many other ways of measuring correlation (e.g., Kendall's Tau; Kendall, 1938).

A.2 Pearson Correlation

The **Pearson correlation** is a well-known metric, especially in the context of evaluating systems with continuous outputs. It was introduced by Francis Galton (Galton, 1877, 1889) and later developed by Karl Pearson (Pearson, 1896), and measures the linear relationship between two random variables (Neter et al., 1996). The metric is well-known for being robust to changes in scale (Shardlow et al., 2021). We define it as follows:

Definition A.2.1 (Pearson Correlation)**.** Let the variables $\mathbf{x} = [x_1, \cdots, x_n]$ and $\mathbf{y} = [y_1, \cdots, y_n]$ be two vectors of size $n \in \mathbb{N}$ (that is, n is the number of observations for each dimension), then

$$r_{\mathbf{xy}} = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2}} \tag{A.1}$$

is the Pearson correlation coefficient between $\mathbf{x}$ and $\mathbf{y}$.

It makes the following assumptions:

(1) that both variables should be normally distributed,

(2) (linearity) that there exists a straight line relationship between the two variables,

(3) (homoscedasticity) that the data is equally distributed about the regression line.

Pearson's correlation is particularly sensitive to the distribution of data, so, following Yannakoudakis et al. (2011) we also report Spearman's correlation which is more robust to outliers Shardlow et al. (2020).

A.3 Spearman Rank

Alternatively, **Spearman's rank correlation**, first introduced by Charles Spearman (Spearman, 1961), is a non-parametric text that is used to measure the monotonic relationship between two variables, not necessarily linearly. We add this metric here because it is more robust to outliers than Pearson's correlation (Shardlow et al., 2020). It is defined in the following way:

Definition A.3.1 (Spearman Rank Correlation)**.** Let the variables $\mathbf{x} = [x_1, \cdots, x_n]$ and $\mathbf{y} = [y_1, \cdots, y_n]$ be two vectors of size $n \in \mathbb{N}$ (that is, n is the number of observations for each dimension), then the Spearman rank correlation is given by

$$\rho_{\mathbf{xy}} = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{A.2}$$

where for all $1 \leq i \leq n \in \mathbb{N}$, d_i is the difference between the **ranks** of x_i and y_i. In our applications, the values $x_1, \cdots, x_n$ and $y_1, \cdots, y_n$ are respectively ranked from highest (rank 1) to lowest (rank n).

Unlike the Pearson correlation, this metric does not carry any assumptions about the distribution of the data. It does however assume that the data must be (1) at least ordinal, and (2) that the scores on one variable must be monotonically related to the other variable.

For both metrics, Cohen's standard can be used to evaluate the correlation coefficient to determine the strength of the relationship, or the **effect size** (Sullivan and Feinn, 2012; Cohen, 1960). In particular, correlation coefficients between 0.10 and 0.29 represent a small association, coefficients between 0.30 and 0.49 represent a medium association, and coefficients of 0.50 and above represent a large association (Zou et al., 2003).

Appendix B

ELLIPSE Marking Guidelines

The following marking guidelines were kindly provided to us by Perpetual Baffour, Research Director at The Learning Agency Lab[1]*, who was involved in the creation of the ELLIPSE dataset in the context of the 2022 "Feedback Prize - English Language Learning" Kaggle competition.*[5] *They most kindly agreed to us using and including the rubric in this report, since, at the time of writing, it was not publicly available anywhere else. Note however that they are planning on releasing this rubric on their website very soon. They are also due to publish a paper describing the corpus and rubric in more detail (Crossley et al., forthcoming) in the near future. An early version of this paper was shared with us for the benefit of our study.*

B.1 Key Terms and Definitions

Definition B.1.1 (Phrase)**.** Multiple word units.

Definition B.1.2 (Grammar)**.** The rules by which words change their forms, including the use of word classes and grammatical morphology in English. Word classes include prepositions, pronouns, nouns, verbs, etc. Grammatical morphology includes third person, plural, possessive, etc.

Definition B.1.3 (Syntax)**.** Structuring sentences according to syntactic rules related to coordinating clauses, developing syntactic phrases (noun, verb, preposition phrases), phrasal and clausal dependency, and transformations such as passives, relative clauses, and negations.

Definition B.1.4 (Cohesive device)**. Cohesive devices** are used as links between two or more items (e.g., words, phrases, clauses) in a text to enhance text cohesion. These include the use of conjunctions ("and", "but", "if", "on the other hand"), transitions ("first", "next", "finally", "for example"), repetition of words, phrases, and ideas across sentences and paragraphs, and the use of anaphora (pronouns replacing nouns).

Definition B.1.5. Sentences can either be

(1) **simple**, independent clause,

(2) **complex**, with independent and dependent clauses, or

(3) **compound**, with two of more independent clauses.

Definition B.1.6 (Chunks)**.** Multiple words that combine to have a single meaning. Often memorised without knowing what the individual words mean (e.g., "How are you" for "Hello").

Definition B.1.7 (Lexical bundles)**.** Multiple word units that are common in English but are not idiomatic ("There is"). More common than **collocations**.

Definition B.1.8 (Collocation)**.** Two or more words that are often used together (e.g., "save time", "go to bed", "fast food")

Definition B.1.9 (Idiom)**.** Multi-word unit where meaning not deducible from those of the individual words (e.g., "kick the bucket" or "rain cats and dogs".)

B.2 Scoring Rubric

	Phraseology	Grammar	Conventions
5	Flexible and effective use of a variety of phrases, such as idioms, collocations, and lexical bundles, to convey precise and subtle meanings; rare minor inaccuracies that are negligible.	Command of grammar and usage with few or no errors.	Consistent use of appropriate conventions to convey meaning; spelling, capitalisation, and punctuation errors nonexistent or negligible.
4	Appropriate use of a variety of phrases, such as idioms, collocations, and lexical bundles; occasional inaccuracies and colloquialisms.	Minimal errors in grammar and usage.	Generally consistent use of appropriate conventions to convey meaning; spelling, capitalisation, and punctuation errors few and not distracting.
3	Evident use of phrases such as idioms, collocations, and lexical bundles but without much variety; some noticeable repetitions and misuses.	Some errors in grammar and usage.	Developing use of conventions to convey meaning; errors in spelling, capitalisation, and punctuation that are sometimes distracting.
2	Narrow range of phrases, such as collocations and lexical bundles, used to convey basic and elementary meaning; many repetitions and/or misuses of phrases.	Many errors in grammar and usage.	Variable use of conventions; spelling, capitalisation, and punctuation errors frequent and distracting.
1	Memorized chunks of language, or simple phrasal patterns predominate; many repetitions and misuses of phrases.	Errors in grammar and usage throughout.	Minimal use of conventions; spelling, capitalisation, and punctuation errors throughout.

	Cohesion	Syntax	Vocabulary	Overall
5	Text organisation consistently well controlled using a variety of effective linguistic features such as reference and transitional words and phrases to connect ideas across sentences and paragraphs; appropriate overlap of ideas.	Flexible and effective use of a full range of syntactic structures including simple, compound, and complex sentences; there may be rare minor and negligible errors in sentence formation.	Wide range of vocabulary flexibly and effectively used to convey precise meanings; skillful use of topic-related terms and less common words; rare negligible inaccuracies in word use.	Native-like facility in the use of language with syntactic variety, appropriate word choice and phrases; well-controlled text organisation; precise use of grammar and conventions; rare language inaccuracies that do not impede communication.
4	Organisation generally well controlled; a range of cohesive devices used appropriately such as reference and transitional words and phrases to connect ideas; generally appropriate overlap of ideas.	Appropriate use of a variety of syntactic structures, such as simple, compound, and complex sentences; occasional errors or inappropriateness in sentence formation.	Sufficient range of vocabulary to allow flexibility and precision; appropriate use of topic-related terms and less common lexical items.	Facility in the use of language with syntactic variety and range of words and phrases; controlled organisation; accuracy in grammar and conventions; occasional language inaccuracies that rarely impede communication.
3	Organisation generally controlled; cohesive devices used but limited in type; some repetitive, mechanical, or faulty use of cohesion use within and/or between sentences and paragraphs.	Simple, compound, and complex syntactic structures present although the range may be limited; some apparent errors in sentence formation, especially in more complex sentences.	Minimally adequate range of vocabulary for the topic; no precise use of subtle word meanings; topic related terms only used occasionally; attempts to use less common vocabulary but with some inaccuracy.	Facility limited to the use of common structures and generic vocabulary; organisation generally controlled although connection sometimes absent or unsuccessful; errors in grammar and syntax and usage. Communication is impeded by language inaccuracies in some cases.
2	Organisation only partially developed with a lack of logical sequencing of ideas; some basic cohesive devices used but with inaccuracy or repetition.	Some sentence variation used; many sentence structure problems.	Narrow range of vocabulary to convey basic and elementary meaning; topic related terms used inappropriately; errors in word formation and word choice that may distort meanings.	Inconsistent facility in sentence formation, word choice, and mechanics; organisation partially developed but may be missing or unsuccessful. Communication impeded in many instances by language inaccuracies.
1	No clear control of organisation; cohesive devices not present or unsuccessfully used; presentation of ideas unclear.	Pervasive and basic errors in sentence structure and word order that cause confusion; basic sentences errors common.	Limited vocabulary often inappropriately used; limited control of word choice and word forms; little attempt to use topic-related terms.	A limited range of familiar words or phrases loosely strung together; frequent errors in grammar (including syntax) and usage. Communication impeded in most cases by language inaccuracies.