Indian Institute of Technology Bombay

# Multi-Task Learning For Automated Essay Scoring

Rahul Kumar,
IIT Patna

Internship Report

*Supervisor*

Prof. Pushpak Bhattacharyya
Department of Computer Science and Engineering,
Indian Institute of Technology, Bombay

July 15, 2019

# Abstract

Multi-Task learning is a sub-field of Machine Learning that aims to solve multiple different tasks at the same time. It does this learning task by taking advantages of similarity between the different tasks.In this report, we demonstrate multi-task learning for automated essay scoring.We explain how we can implement multi-task learning in automatic essay assessment.We choose feature-less-engineering method because manually grading student's essays is labor intensive.We choose a suitable model as a baseline, we implement it from scratch, tune it for GloVe word embeddings and train it on ASAP++ datasets. We then introduce multi-task learning on top of baseline model.We further improve the model by giving input the attributes scores and essay representation to fully connected layer with sigmoid activation and discuss the reasons for this choice.

# Acknowledgements

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to my advisor, Prof. Pushpak Bhattacharyya whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project especially in writing this report.

Many thanks go to the head of the project, and my mentor Sandeep Mathias who have invested his full effort in guiding me in achieving the goal.I would like to thank my parents who always keep me motivated.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Introduction and Motivation

Writing skills are always essential for success in school, career, and society. Revision and feedback are an important for creating good writing material. Students in general require input from their teachers for mastering the art of writing. Giving feedback to a large group of students on frequent writing assignments can be pretty hectic from the teachers point of view. Therefore, we are creating a system that can be accurate in both providing feedback as well as grading performance in essay writing that can also be time efficient.

This project aims to build a machine learning system for automatic scoring of essays written by students. The basic idea is to build a multi-task learning system which can model the attributes like language fluency, vocabulary, structure, organization, content, etc. simultaneously along with calculating the overall score of the essay.

## 1.2 Problem Statement

Our aim is to build a model that can take an essay as input and automatically output the score of that essay along with the scores of its attributes, like content, organization, language, etc. Within the scope of this project, we only work with essays written by students ranging in grade levels from Grade 7 to Grade 10. We wanted to avoid feature engineering so that our system can be easily used for different types of essay data.

## 1.3 The ASAP Datasets

We use the dataset[1] provided for Hewlett Foundations Automated Student Assessment Prize (ASAP) competition on Kaggle. Some characteristics of the dataset:

1. There are 8 different sets of essays, each essay ranges from an average length of 150 to 650 words per response.

2. Each essay was graded by two or three instructors.

[1] The dataset can be downloaded from here: https://www.kaggle.com/c/asap-sas/data/

3. Each set has a different grading scale.

Table 1.1 presents an overview of the datasets.

| Prompt Id | Essay Type | Avg. Length | No. of Essays | Score Range |
|---|---|---|---|---|
| Prompt 1 | Argumentative | 1785 | 350 | 1 - 6 |
| Prompt 2 | Argumentative | 1800 | 350 | 1 - 6 |
| Prompt 3 | Source-Dependent | 1726 | 150 | 0 - 3 |
| Prompt 4 | Source-Dependent | 1772 | 150 | 0 - 3 |
| Prompt 5 | Source-Dependent | 1805 | 150 | 0 - 4 |
| Prompt 6 | Source-Dependent | 1800 | 150 | 0 - 4 |
| Prompt 7 | Narrative | 1569 | 300 | 0 - 3 |
| Prompt 8 | Narrative | 723 | 650 | 1 - 6 |

**Table 1.1:** *ASAP automatic essay grading datasets statistics.*

### 1.3.1 Types of Essays

1. **Argumentative/ Persuasive essays** - These are essays where the prompt is one in which the writer has to convince the reader about their stance for or against a topic (for example, free speech in public colleges).

2. **Source-dependent responses** - These essays are responses to a source text, where the writer responds to a question about the text (for instance, describing the writers opinion about an incident that happened to him in the text).

3. **Narrative/ Descriptive essays** - These are essays where the prompt requires us to describe / narrate a story.

### 1.3.2 Attributes of Argumentative / Persuasive Essays

There are 5 attributes for narrative essays, namely

1. **Content**: The quantity of relevant text present in the essay.

2. **Organization**: The way the essay is structured.

3. **Word Choice**: The choice and aptness of the vocabulary used in the essay.

4. **Sentence Fluency**: The quality of the sentences in the essay.

5. **Conventions**: Overall writing conventions to be followed, like spelling, punctuations, etc.

### 1.3.3 Attributes of Source-dependent Responses

There are 4 attributes for source-dependent responses, namely

1. **Content**: The amount of relevant text present in the essay.

2. **Prompt Adherence**: A measure of how the writer sticks to the question asked in the prompt.
3. **Language**: The quality of the grammar and spelling in the response.
4. **Narrativity**: A measure of the coherence and cohesion of the response to the prompt.

These are the attributes which we can use while training the multi-task learning model.

## 1.4 Our Contributions

Most automated essay scoring systems are based on hand-crafted features and supervised machine learning algorithms. Although these models performs well to some extent, they fail to capture those additional information which are not added in the model. To address this problem, we propose a novel approach based on multi-task learning over recurrent neural networks, which learns the features automatically from essay texts. We show that our system outperforms existing essay scoring systems, without using any manually designed features.

Our proposed neural essay scoring system assigns a holistic score to a given essay based on its writing quality. However, to be useful for language learners, an AES system should provide further feedback to the users to help improve their writing skill. Such feedback may include some information about the organization of the essay, the strength of the argument, grammar mistakes, etc.

## 1.5 Summary

This chapter gives a brief introduction about this report.We also provided the description of ASAP datasets.

The report has a literature survey in Chapter 2. The literature survey covers a description of the topic and limitations of systems including recent work done in using deep learning to score an essay. We describe our systems in Chapters 3 and 4. We conclude our report and describe future work in Chapter 5.

# Chapter 2

# Literature Survey

Essay writing is usually a part of the student assessment process and several educational organizations evaluate the writing skills of students in their examinations. The written essay is then submitted to be assessed by trained human evaluators. Because of the large number of students participating in these exams, grading all essays is very time-consuming and costly. As a result, evaluating student essays automatically by the help of computer programs has attracted the attention of many researchers during the past 50 years (Page, [8], [9]; Shermis and Burstein, [17]; Attali and Burstein, [2]; Shermis and Burstein, [16]; Shermis and Hamner, [18]). Moreover, such computer programs can be used to provide cheap accessible educational services to language learners all over the world. Therefore, automated essay scoring programs have made an impact on the education sector globally.

In this chapter, we first briefly discuss the writing aspects that are considered by human evaluators for assessing the quality of student essays. Next, we give a more detailed description of the task that we are addressing in this report, and we discuss the advantages and limitations of the existing essay scoring systems. Finally, we will summarize our contributions in this report.

## 2.1 Essay Evaluation Criteria

An ideal Automated Essay evaluation system would provide feedback along with the overall score based on the quality of the written essays according to the various essay evaluation criteria. Human evaluators consider several criteria for essay evaluation. Some of the most important criteria are mentioned below:

- **Essay organization:** Whether the essay is structured properly, develops the main idea logically, and is well organized.
- **Prompt adherence**: Whether the essay is addressing the given prompt and stays on-topic throughout the essay.
- **Argument strength:** Whether the argument made in the essay would convince the reader.

- **Essay length:** Whether the number of words in the given essay is within the accepted range of the examination.
- **Textual errors**: Whether the essay is free from grammatical errors, spelling errors, punctuation errors, etc.
- **Word choice:** Whether the writer has used the correct and appropriate set of words throughout the essay.
- **Readability:** Whether the essay is easy to read.
- **Coherence:** Whether or not the reader can follow the essay easily.

Therefore, an automated essay evaluation system should ideally provide some feedback about the different aspects mentioned above, in addition to assigning a overall score to the given essay. Such feedback can help students to improve their writing skills and learn about the key aspects of essay writing. The systems that provide overall scores along with some feedbacks are called automated essay scoring (AES) systems.

In this report, we focus on automated essay scoring systems and will address several problems in this field. We first provide a description of the automated essay scoring task and discuss some advantages of this system. We then discuss some of the limitations of existing AES systems.

## 2.2 Automated Essay Scoring

Automated Essay Scoring refers to the process of assigning score to the student essays without human interference. It has the potential to reduce the preprocessing costs, speed up the reporting results and improve the consistency of scoring.An AES system takes as input an essay written for a particular prompt, and then assigns a numeric score to the essay reflecting its quality. Since the output of an AES system for each essay is usually a real-valued number, the task is often addressed as a supervised machine learning task (mostly by regression or preference ranking) and learning algorithms are used to discover the relationship between essays and their reference scores. Since human graders can easily recognize essays by various styles of writing and creativity, writer's ability of thinking and evaluates the correctness of the writings. However, for several reasons, human raters can also be inefficient, can do some mistakes in scoring process. Some of the human errors are mentioned below:

- **Severity and leniency:** Some human raters consistently assign higher scores to essays, while some others give lower scores, even when evaluating the same aspects of writing.
- **Scale shrinkage:** Extreme categories on a scale are not used by some human raters.
- **Inconsistency:** Since writing aspects and the rubric are not well-defined concepts, human raters usually have different understandings of these dimensions, and therefore, do not assess essays consistently.

- **Stereotyping:** Human raters are vulnerable to erratic judgment of individuals, based on predetermined impressions about them.
- **Perception difference:** Some human raters evaluate essays erratically, because of their past grading experiences.
- **Rater drift:** Human raters usually assess essays inconsistently over time.

On the other hand, automated essay scoring systems are not affected by these sources of errors. Although these systems may learn to assign erratic scores to essays because of erroneous training data (e.g., affected by severity or leniency), automated essay scoring systems are fast and efficient, consistent, loyal to the programmed definitions and learned patterns from data, and not vulnerable to stereotyping, perception difference, and rater drift.

## 2.3 Related Work

Most of the automated essay scoring engines used some kind of supervised machine learning techniques, with or without manual feature engineering.

Isaac Persing and Vincent Ng worked on modelling essay qualities like stance [14], argument strength [13], prompt adherence [12], thesis clarity [11], and organization [10] on International Corpus of Learner English (ICLE) Dataset. They use feature-rich machine learning regression techniques for scoring each quality, where the features are extracted based on a set of rules.

Kaveh Taghipour and Hwee Tou Ng [19] use a recurrent neural network approach for essay scoring to bypass feature engineering. They achieve a significant improvement over a strong baseline in terms of QWK. This method thus may be applied to a variety of datasets without the explicit and laborious process of feature engineering.

Ronan Cummins and Marek Rei [4] try to simultaneously solve two tasks in automated assessment namely, grammatical error detection and automated essay scoring, using a multi-task neural network that jointly optimizes for both tasks. They show that the task of automated essay scoring can be significantly improved in this way.

## 2.4 Limitation of AES

Since we have seen some major advantages of an AES system but they also has some limitations and there are some concerns about using these systems for student writing evaluation. Some of the limitations are as follows:

- **Agreement with human raters:** There is still a gap between the performance of human raters and an AES system and that should be improved. Many researchers are currently trying to improve the performance of AES systems.
- **Modeling writing traits:** Although many existing automated essay scoring systems try to capture some information about high level aspects of essay writing the systems that do model these traits are still inaccurate.

- **Robustness:** Automated essay scoring systems can be gamed by certain types of nonsensical essays, generated by a computer program or written by a human.
- **Detailed feedback:** For an AES system to be helpful in self-learning situations, it has to provide proper feedback about word choices, textual errors, and content of an essay. Such feedback can be utilized by language learners to improve their essays and learn the required aspects of writing quickly.
- **Generalization to unseen styles of writing:** Since AES systems do not use the same cognitive processes as the human brain, they may not be able to make accurate judgments about the quality of essays with unique writing styles. Therefore, these systems may not make valid and fair predictions about such writing styles.

The limitations of automated essay scoring systems have raised questions and concerns about the usage of these systems.Therefore, researchers have tried to tackle these issues in the past and have made significant progress in this field. However, we are still far from an ideal automated essay scoring system that can replace human raters completely.We will address some of the limitations mentioned above in this report. More specifically, we propose a new approach to essay scoring to achieve better agreement with human raters (first limitation), develop systems to model essay writing traits (second limitation), and design a module to make automated essay scoring systems robust against computer-generated essays (third limitation).

## 2.5 The Use of Deep Learning

Many tasks in NLP are starting to move towards using deep learning for solutions, ever since the publication of Collobert et al.'s 2011 work - *Natural Language Processing (Almost) from Scratch* - on using neural networks to solve part-of-speech tagging, chunking, named-entity recognition, and semantic role labeling[3].

In 2012, a dataset released by Kaggle - the Automatic Student Assessment Prize (ASAP) Automatic Essay Grading (AEG) dataset[1] - became the go-to dataset for essay grading. It has a total of about 13,000 essays which have their scores labeled across 8 essay prompts. This dataset has inspired a lot of the recent work done on essay grading. In this section, we discuss a few specific works done on essay grading using deep-learning.

For the 2016 edition of the ACL, Alikaniotis et al. [1] came up with a way to generate word embeddings based on the different essay scores.

Later on, at EMNLP 2016, Taghipour and Ng [19] and Dong and Zhang [5] presented deep learning techniques to predict the overall score of essays. Taghipour and Ng [19] used LSTMs (Long Short-Term Memory), while Dong and Zhang used CNNs (Convolution Neural Networks) in their solutions.

[1] The dataset can be downloaded from here: https://www.kaggle.com/c/asap-sas/data/

Another field of study that came out of the ASAP dataset was the area of cross-domain essay grading, in which you train on one prompt and test on another prompt. The earliest work on this was from Phandi et al. [15] at EMNLP 2015. Dong and Zhang [5] showed how using deep learning can aid in this task as well at EMNLP 2016. The most recent work in essay grading has also come from Dong and Zhang [6] at CoNLL 2017.

## 2.6 Summary

In this chapter, we review the literature necessary for our report. We looked at different literature on using deep learning and limitations of AES system.

# Chapter 3

# Implementation

## 3.1 Our approach

Our model is inspired by hierarchical neural network in these papers using attention [6] and co-attention [20]. In the self-attention model [7], they considered each essay as a sequence of sentences rather than a sequence of words. Their model has three parts. First, they used a convolution layer and attention pooling layer to get sentence representation. Second, they used an LSTM layer and another attention pooling layer for document representation. Finally, they used a sigmoid layer for score prediction. whereas co-attention model [20] they considered bidirectional attention flow layer for document representation and an additional modelling layer. By doing so, they can capture the relationships between the student essays and source dependent article. In particular, a higher attention score will be assigned to sentences that are mentioned in the article but less mentioned in other essays.

Our model is a simple and natural extension to these models where we are considering traits score provided by ASAP++ datasets. As it contains scores for different attributes like content, organization, fluency, etc. So, we have designed a multi-task learning model on top of these baseline model [20] [7]. We the added dense layer for collecting attributes scores and finally considered one fully connected layer with sigmoid activation for overall score and measure the performance. Figure: 3.1 shows the baseline model for self-attention and Figure: 3.2 shows the baseline model for co-attention.

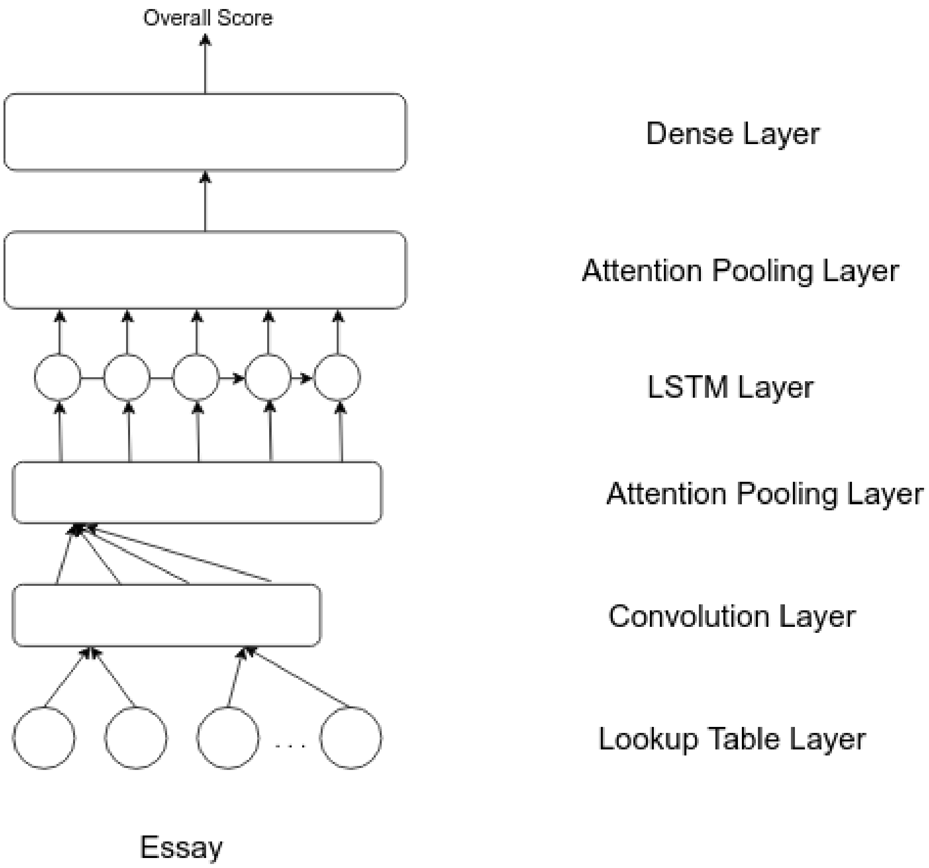

Figure 3.1: *Baseline Model for self-attn*

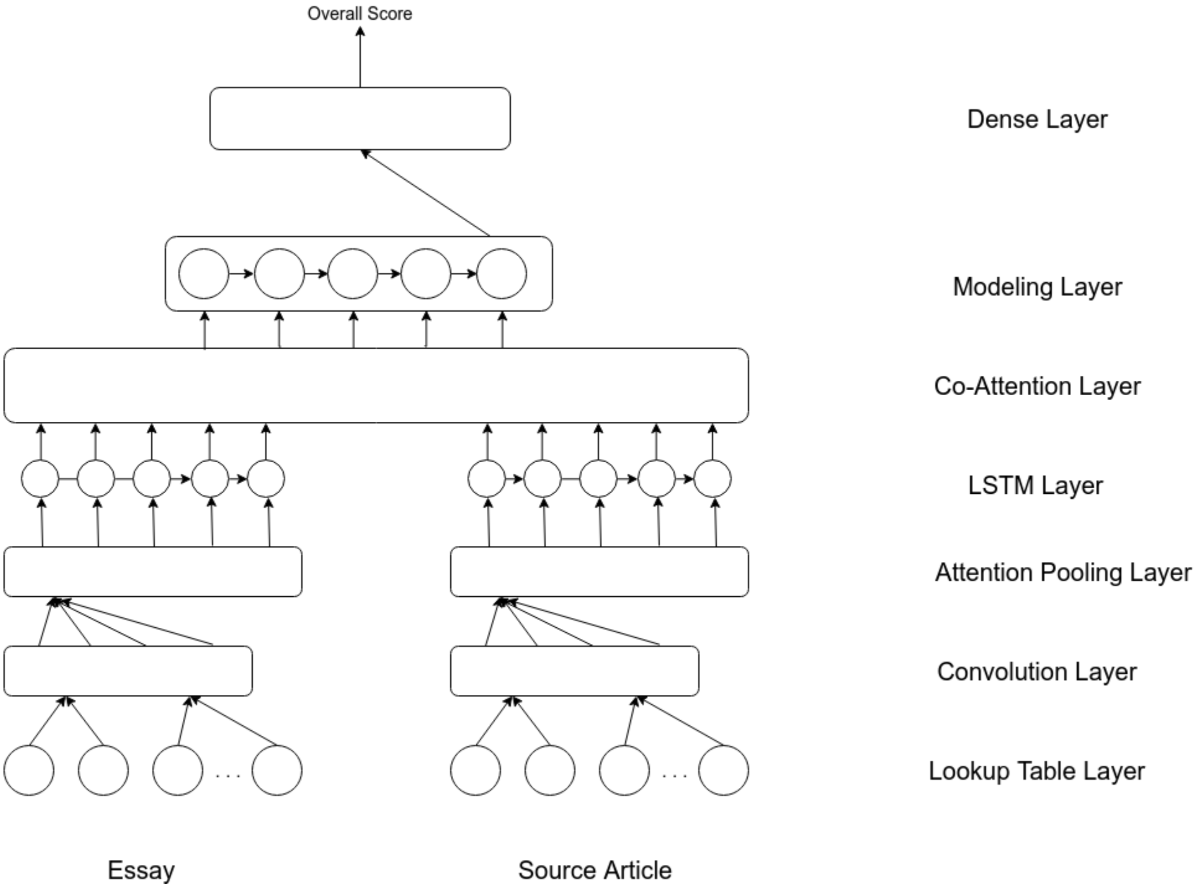

Figure 3.2: *Baseline Model for co-attn*

## 3.2 Baseline Model

Baseline model is a neural network made up of the following layers:

1. Lookup Table Layer (Input)
2. Convolution Layer
3. Attention Pooling Layer
4. Recurrent Layer
5. Sentence Level Co-Attention Layer
6. Modelling Layer for Co-Attention
7. Linear Layer with Sigmoid Activation (Output)

## 3.3 Data pre-processing

First a tokenizer is generated on a large set of essays. The tokenizer assigns a unique positive integer for every word in the essay set. The number zero is not given to any word, as it will be used later to represent spelling mistake/unknown words.

Given a tokenizer and an input essay, first the essay is converted to lower case, then each word in essay is replaced by its unique representative number by tokenizer. As a neural network only accepts input of a fixed shape and dimension, an essay is either (post) padded with zeros or truncated to a fixed length. Now an integer encoding of essay is obtained, which can be fed to a neural network.

## 3.4 Word embeddings

This layer maps each word in sentences to a high dimension vector. The purpose of a word embedding is to essentially capture all the possible meaning and semantics of the corresponding word. That means that in the word embedding space, semantically close words are closer than semantically unrelated words.

We use the GloVe pre-trained word embeddings to obtain the word embedding vector for each word. It was trained on 6 billion words from Wikipedia 2014 and Gigaword 5. It has 400,000 uncased vocabulary items. The dimensionality of GloVe in our model is 50 dimensions. Given the tokenizer and word embeddings, we created an embeddings matrix **E** where the for every word W indexed by tokenizer with an index I, the $\text{I}^{\text{th}}$ row contains the embedding vector of the word W, if the word embeddings contains word W, else $\text{I}^{\text{th}}$ row has all zeros representing unknown word.

## 3.5 Lookup Table Layer

The first layer of our neural network projects each word into a $\mathrm{d}_{\mathrm{LT}}$ dimensional space. Given a sequence of words $W$ represented by their one-hot representations ($\mathbf{w}_1$ ,$\mathbf{w}_2$ ,. . . , $\mathbf{w}_{\mathrm{M}}$ ), the output of the lookup table layer is calculated by this Equation:

$$LT(W) = (\mathbf{E.w}_1 \ , \mathbf{E.w}_2 \ , \ . \ . \ ., \mathbf{E.w}_{\mathrm{M}})$$

**E** is the word embeddings matrix and will be learned during training.The performance of the system is dependent upon the word embedding system.

## 3.6 Convolution Layer

Once the dense representation of the input sequence $W$ is calculated, it is fed into the recurrent layer of the network. However, it might be beneficial for the network to extract local features from the sequence before applying the recurrent operation. This optional characteristic can be achieved by applying a convolution layer on the output of the lookup table layer. In order to extract local features from the sequence, the convolution layer applies a linear transformation to all $M$ windows in the given sequence of vectors.Given a window of dense word representations **x1**, **x2** , . . ., **xl** , the convolution layer first concatenates these vectors to form a vector $\bar{x}$ of length $l.d_{\mathrm{LT}}$ and then uses this Equation to calculate the output vector of length $d_{\mathrm{c}}$ :

$$Conv(\bar{\mathbf{x}}) = \mathbf{W}.\bar{\mathbf{x}} + \mathbf{b}$$

where **W** and **b** are the parameters of the network and are shared across all windows in the sequence. The convolution layer can be seen as a function that extracts feature vectors from $n$-grams. Since this layer provides $n$-gram level information to the subsequent layers of the neural network, it can potentially capture local contextual dependencies in the essay and consequently improve the performance of the system.

## 3.7 Attention Pooling Layer

After Convolution layer, they added this layer for sentence representations. The attention layer is defined by these equations:

$$m_{\mathrm{i}} = \tanh(U_{\mathrm{m}}.p_{\mathrm{i}} + b_{\mathrm{m}})$$

$$v_{\mathrm{i}} = \frac{e^{u_v \cdot m_i}}{\sum e^{u_v \cdot m_i}}$$

$$s = \sum v_{\mathrm{i}} p_{\mathrm{i}}$$

where $U_{\mathrm{m}}$ ,$u_{\mathrm{v}}$ and $b_{\mathrm{m}}$ are weight matrix, vector, and bias vector, respectively. $m_{\mathrm{i}}$ and $v_{\mathrm{i}}$ are attention vector and attention weight for $p_{\mathrm{i}}$.

## 3.8 Recurrent Layer

After generating the sentence representations, we use Long Short term memory (LSTM) to capture contextual evidence from previous sentences to refine the sentence representation. This representation should ideally encode all the information required for grading the essay. However, since the essays are usually long, consisting of hundreds of words, the learned vector representation might not be sufficient for accurate scoring. For this reason, we preserve all the intermediate states of the recurrent layer to keep track of the important bits of information from processing the essay.

Long short-term memory units are modified recurrent units that can cope with the problem of vanishing gradients more effectively. LSTMs can learn to preserve or forget the information required for the final representation. In order to control the flow of information during processing of the input sequence, LSTM units make use of three gates to discard (forget) or pass the information through time. The following equations formally describe the LSTM function:

$$
\begin{gathered}
\mathbf{i}_{\mathrm{t}} = \sigma(\mathbf{W}_{\mathrm{i}}.\mathbf{x}_{\mathrm{t}} + \mathbf{U}_{\mathrm{i}}.\mathbf{h}_{\mathrm{t-1}} + \mathbf{b}_{\mathrm{i}}) \\
\mathbf{f}_{\mathrm{t}} = \sigma(\mathbf{W}_{\mathrm{f}}.\mathbf{x}_{\mathrm{t}} + \mathbf{U}_{\mathrm{f}}.\mathbf{h}_{\mathrm{t-1}} + \mathbf{b}_{\mathrm{f}}) \\
\widetilde{\mathbf{c}} = \tanh(\mathbf{W}_{\mathrm{c}}.\mathbf{x}_{\mathrm{t}} + \mathbf{U}_{\mathrm{c}}.\mathbf{h}_{\mathrm{t-1}} + \mathbf{b}_{\mathrm{c}}) \\
\mathbf{c}_{\mathrm{t}} = \mathbf{i}_{\mathrm{t}}\, o\, \widetilde{\mathbf{c}} + \mathbf{f}_{\mathrm{t}}\, o\ \mathbf{c}_{\mathrm{t-1}} \\
\mathbf{o}_{\mathrm{t}} = \sigma(\mathbf{W}_{\mathrm{o}}.\mathbf{x}_{\mathrm{t}} + \mathbf{U}_{\mathrm{o}}.\mathbf{h}_{\mathrm{t-1}} + \mathbf{b}_{\mathrm{o}}) \\
\mathbf{h}_{\mathrm{t}} = \mathbf{o}_{\mathrm{t}}\, o \tanh(\ \mathbf{c}_{\mathrm{t}})
\end{gathered}
$$

$x_{\mathrm{t}}$ and $h_{\mathrm{t}}$ are the input and output vectors at time t, respectively. $\mathbf{W}_{\mathrm{i}}$ , $\mathbf{W}_{\mathrm{f}}$ , $\mathbf{W}_{\mathrm{c}}$ , $\mathbf{W}_{\mathrm{o}}$ , $\mathbf{U}_{\mathrm{i}}$ , $\mathbf{U}_{\mathrm{f}}$ , $\mathbf{U}_{\mathrm{c}}$ , and $\mathbf{U}_{\mathrm{o}}$ are weight matrices and $\mathbf{b}_{\mathrm{i}}$ , $\mathbf{b}_{\mathrm{f}}$ , $\mathbf{b}_{\mathrm{c}}$ , and $\mathbf{b}_{\mathrm{o}}$ are bias vectors. The symbol $o$ denotes element-wise multiplication and $\sigma$ represents the sigmoid function.

## 3.9 Sentence Level Co-Attention Layer

We are using this layer only in this co-attn based model[co-attn paper]. This layer links the information between essays and source dependent article and generates a collections of aware features vector of essay sentences.

## 3.10 Modelling Layer for Co-Attention

Now we have the sentence representations of essay. So, we need one more layer to represent essay representations. Therefore, we introduce another LSTM layer for modeling the essay and only use the output of the final LSTM unit as the output of this layer.

## 3.11 Linear Layer with Sigmoid Activation

After obtaining the essay representation M , a linear layer with sigmoid activation will predict the final output.It is a linear transformation of the input vector and therefore, the computed value is not bounded. Since we need a bounded value in the range of valid scores for each prompt, we apply a sigmoid function to limit the possible scores to the range of (0, 1). The mapping of the linear layer after applying the sigmoid activation function is given by this Equation

$$\mathbf{y} = sigmoid(\mathbf{W}_{\mathrm{o}}\mathbf{M} + \mathbf{b}_{\mathrm{o}})$$

where $\mathbf{W}_{\mathrm{o}}$ is weight vector, and $\mathbf{b}_{\mathrm{o}}$ is bias vector. y is the final predicted score of the essay.

We normalize all gold-standard scores to [0, 1] and use them to train the network. However, during testing, we rescale the output of the network to the original score range and use the rescaled scores to evaluate the system.

## 3.12 Evaluation Metric

Quadratic weighted kappa is a metric to measure accuracy of an essay scoring system. It takes as input the true scores and predicted score and outputs a value between negative infinity to one, one being fully correlated and as a consequence corresponds to 100% accuracy. We evaluated our performance with the quadratic weighted kappa metric defined below:

$$\kappa = 1 - \frac{\sum_{i=1}^{k}\sum_{j=1}^{k} O_{ij}W_{ij}}{\sum_{i=1}^{k}\sum_{j=1}^{k} E_{ij}W_{ij}}$$

where $\kappa$ is the QWK, $k$ is the number of distinct grades for that aspect, $O_{ij}$, $E_{ij}$ and $W_{ij}$ are the values stored in the observation, expected and weight matrices respectively.

The Weight matrix W is calculated based on the difference between raters scores. Let the scores range from 1, 2 .. , N:

$$W_{\text{ij}} = 1 - \tfrac{(i-j)^2}{(N-1)^2},$$

where $W$ is the weights matrix, $N$ is the number of classes.
The Quadratic Weighted Kappa (QWK) metric typically varies from 0 - only random agreement between raters - to 1 (complete agreement between raters). In the event that there is less agreement between the raters than expected by chance, this metric may go below 0.

## 3.13 Experiments

In the later sections, we describe our experimental setup and present the results. Moreover, an analysis of the results and some discussion are provided in this section.

## 3.14 Setup

The dataset that we have used in our experiments is the same dataset used in the ASAP competition run by Kaggle (see Table 1 for some statistics). We use quadratic weighted Kappa as the evaluation metric, following the ASAP competition. Since the test set used in the competition is not publicly available, we use 5-fold cross validation to evaluate our systems. In each fold, 60% of the data is used as our training set, 20% as the development set, and 20% as the test set. We train the model for a fixed number of epochs and then choose the best model based on the development set.

We regularized the network by using dropout and we set the dropout probability to 0.1. Finally, we initialized the lookup table layer using pre-trained GloVe word embeddings. The hyper parameters were tuned for prompts 1 - 6 and then were kept same across all folds.

## 3.15 Results and Discussion

| Prompts ID | Self-attn | | Co-attn | |
|---|---|---|---|---|
| | Proposed | Our Baseline | Proposed | Our Baseline |
| 1 | 0.822 | 0.812 | – | – |
| 2 | 0.682 | 0.666 | – | – |
| 3 | 0.672 | 0.623 | 0.697 | 0.664 |
| 4 | 0.814 | 0.790 | 0.809 | 0.793 |
| 5 | 0.803 | 0.777 | 0.815 | 0.785 |
| 6 | 0.811 | 0.798 | 0.812 | 0.806 |
| 7 | 0.801 | 0.784 | – | – |
| 8 | 0.705 | 0.675 | – | – |

**Table 3.1:** *QWK performance comparison of proposed vs our baseline*

Table 3.1 shows the baseline average QWK performance value of self-attn and co-attn. We are using prompt 1-8 for self-attn whereas prompt 3-6 for co-attn and for rest, source dependent files are not available.

## 3.16 Summary

In this chapter we looked at the baseline models of self-attn and co-attn and their behaviour with different prompts. In next chapter 4, we introduced about multi-task learning based two models and analyze their performance.

# Chapter 4

# Introducing Multi-Task Learning

In Ronan Cummins and Marek Rei [4] paper, they worked on the two problem statement one is to improve essay scoring and another one is to give information about grammatical error detection. But they use same neural network for both the tasks. They use a linear combination of individual loss functions, with co-efficient summing up to one and demonstrate a good performance improvement in essay scoring task.

About ASAP++ datasets, it contains scores of different attributes like content, organization, fluency, language etc.. These existence of the gold scores for these attributes of the essay give motivation to apply the same technique used by Ronan Cummins and Marek Rei [7] as grammatical error rate is in some sense also an attribute of the essay. With this motivation we designed two multi-task learning models as an extension of our baseline.

## 4.1 MTL Model Using Self-Attention Network

Figure 4.1 shows the architecture of the model. This model is an extension to the baseline as we have M outputs where M = number of attributes of essay for which gold scores exist, from output layer instead of one. This model takes only essay as input whereas second model takes context input as well which we discuss later in this chapter. After getting the attributes scores we passed it into another layer along with essay representations to a fully connected layer with sigmoid activation for overall score. The loss function used is a linear combination of individual loss functions with appropriate weights. These weights are tunable, not trainable.

We used a loss weight of 0.1 for attribute score loss function and 1.0 weight for overall score loss function. The reasons for this choice of weights are enumerated as follows:

1. We have no prior information regarding how much an attribute contributes to overall score. So we gave equal weights to all of them.
2. This was a conscious choice which indicates that more priority must be given to minimize the overall score than individual attribute scores.

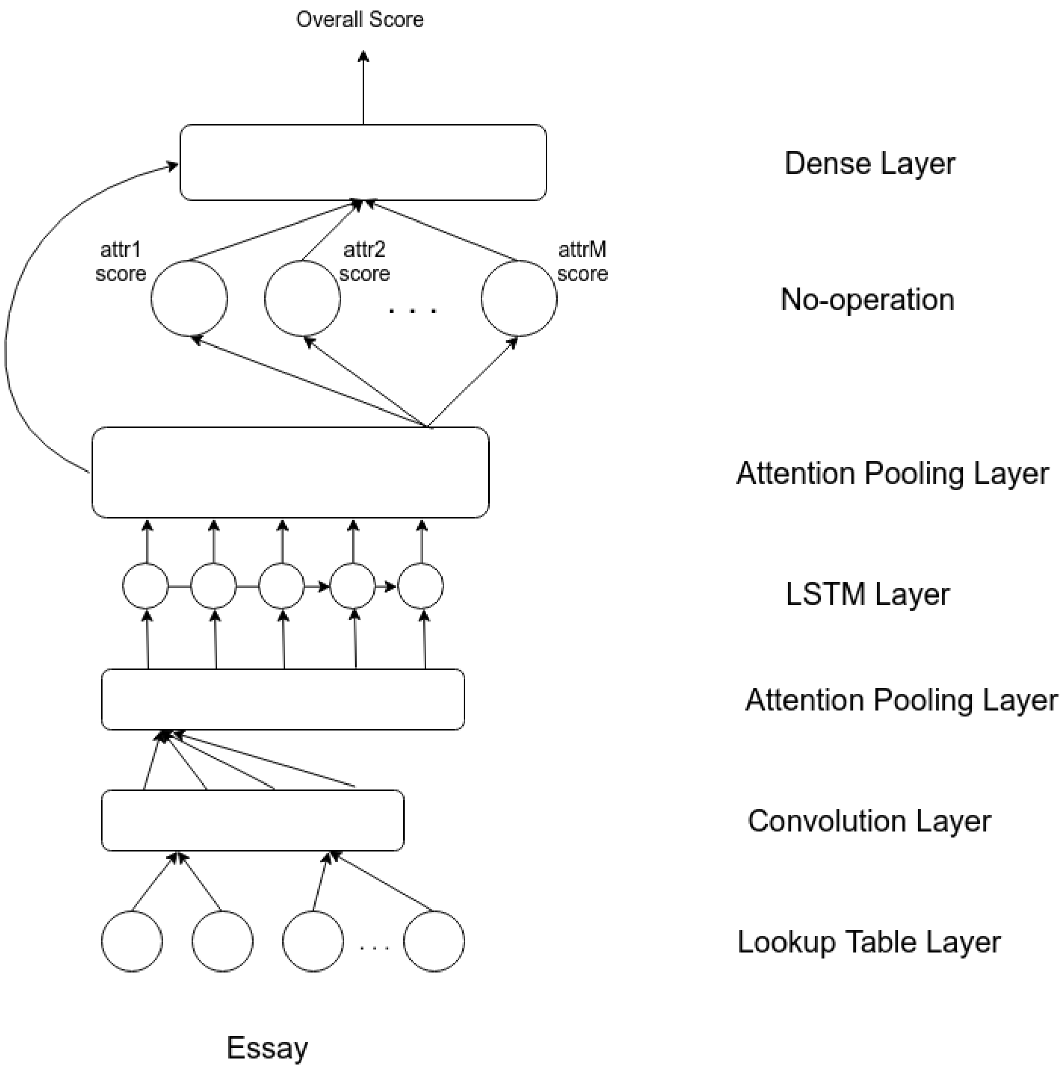

Figure 4.1: *Multi-Task Learning Model 1*

### 4.1.1 Experiments and Results

We tested the model across all prompts from 1-8 in which 1,2 has 5 attributes, 3 to 7 has 4 attributes and 8 has 6 attributes. The results are mentioned below

| S.No. | System | Prompts ID | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| I. | Baseline(Self-attn) | 0.812 | 0.666 | 0.623 | 0.790 | 0.777 | 0.798 | 0.784 | 0.675 |
| II. | Our model | 0.766 | 0.628 | 0.642 | 0.719 | 0.716 | 0.737 | 0.759 | 0.622 |

**Table 4.1:** *QWK performance comparison of baseline vs our model*

## 4.2 MTL Model Using Co-Attention Network

Figure 4.2 shows the architecture of second MTL model. Since prompt 3 to 6 all are source dependent essays which we mentioned in the chapter 2.The difference between this model and the first model is the inputs that we are providing in the lookup table layer. So, we are using essay as well as context as input in this model and getting attribute scores along with an overall score.

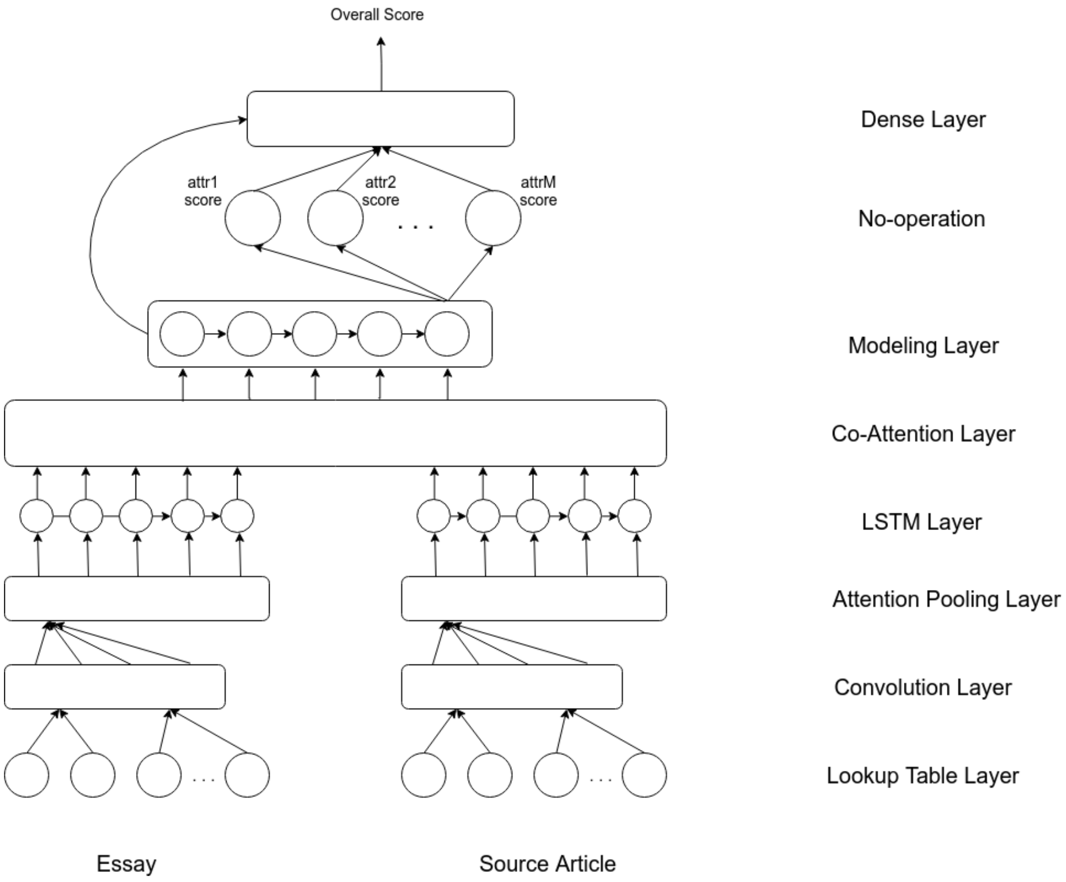

Figure 4.2: *Multi-Task Learning Model 2*

### 4.2.1 Experiments and Results

The setup for all the experiments done in this chapter is same as baseline setup.

| S.No. | System | Prompts ID | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| I. | Baseline(Co-attn) | - | - | 0.664 | 0.793 | 0.785 | 0.806 | - | - |
| II. | Our model | - | - | 0.632 | 0.709 | 0.685 | 0.384 | - | - |

**Table 4.2:** *QWK performance comparison of baseline vs our model*

If we compare Table 4.1 with Table 4.2, the result of self-attn is actually better than co-attn.

## 4.3 Summary

This chapter is all about our approach to multi-task learning. In next chapter 5, we have concluded the report and also explained the reason why we are getting low values in case of co-attn.

# Chapter 5

# Conclusions and Future Work

Automatic essay assessment is an important NLP task in academia. The task is generally done by supervised machine learning techniques that may or may not involve explicit feature engineering. We chose a feature-engineering-less method to ensure portability of our model across various essay datasets. We use GloVe word embeddings in lookup table layer and trained it on ASAP++ competition dataset. We then introduced multi-task learning using a linear combination of overall and attribute loss functions as the neural networks loss function to create our first MTL model without using any source dependent context and finally we compared the results of both the models.

We can improve the model by choosing an appropriate attribute loss weights. Since the second model is complex than the first one which might be one of the reason for getting low values.

# Bibliography

[1] Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. Automatic text scoring using neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 715–725, Berlin, Germany, August 2016. Association for Computational Linguistics.

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

[3] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[4] Ronan Cummins and Marek Rei. Neural multi-task learning in automated assessment. *CoRR*, abs/1801.06830, 2018.

[5] Fei Dong and Yue Zhang. Automatic features for essay scoring – an empirical study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1072–1077, Austin, Texas, November 2016. Association for Computational Linguistics.

[6] Fei Dong, Yue Zhang, and Jie Yang. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[7] Fei Dong, Yue Zhang, and Jie Yang. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162, Vancouver, Canada, August 2017. Association for Computational Linguistics.

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

[9] Ellis B Page. The use of the computer in analyzing student essays. *International review of education*, 14(2):210–225, 1968.

[10] Isaac Persing, Alan Davis, and Vincent Ng. Modeling organization in student essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 229–239, Cambridge, MA, October 2010. Association for Computational Linguistics.

[11] Isaac Persing and Vincent Ng. Modeling thesis clarity in student essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 260–269, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[12] Isaac Persing and Vincent Ng. Modeling prompt adherence in student essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1534–1543, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

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

[14] Isaac Persing and Vincent Ng. Modeling stance in student essays. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2174–2184, Berlin, Germany, August 2016. Association for Computational Linguistics.

[15] Peter Phandi, Kian Ming A. Chai, and Hwee Tou Ng. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[16] Mark D Shermis and Jill Burstein. *Handbook of automated essay evaluation: Current applications and new directions*. Routledge, 2013.

[17] Mark D Shermis and Jill C Burstein. *Automated essay scoring: A cross-disciplinary perspective*. Routledge, 2003.

[18] Mark D Shermis and Ben Hamner. 19 contrasting state-of-the-art automated scoring of essays. *Handbook of automated essay evaluation: Current applications and new directions*, page 313, 2013.

[19] Kaveh Taghipour and Hwee Tou Ng. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas, November 2016. Association for Computational Linguistics.

[20] Haoran Zhang and Diane Litman. Co-attention based neural network for source-dependent essay scoring. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 399–409, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.