Using Automated Essay Evaluation Systems to provide feedback on a Chemistry Abstracting Exercise

Oscar Morris

1 Introduction

An Automated Essay Evaluation (AEE) system is a computational system that can provide a mark and either textual feedback or a breakdown of that mark. The AEE task can be broken down into two parts: 1. Automated Essay Scoring (AES) to provide a grade 2. Automated Feedback Generation (AFG) to provide textual feedback or a breakdown of the mark. AES systems have been well researched and are capable of grading essays almost within the level of disagreement between human graders. AFG systems on the other hand, have been very scarcely researched which may be due to a distinct lack of available datasets for the task. However, developments in pre-trained text summarisation models are likely to perform well at AFG if fine-tuned to the task using a sufficiently large dataset.

Many exam-like exercises such as the abstracting exercise demonstrated in this paper contain questions with a definite correct answer, such as the correctly formatted reference for a specific paper, or the Journal's name. The questions do not need a machine learning approach, significantly decreasing the necessary compute time and increasing the accuracy of the scoring system.

2 Dataset

The dataset used for these experiments was derived from a second-year chemistry abstracting exercise where each tutorial group was given a paper, on varying topics, for which they had to write an abstract and provide other information. The information the students had to provide is as follows:

1. The “Impact Factor” of the Journal
2. The Reference in the Royal Society of Chemistry format
3. The Reference in the American Chemical Society format
4. The number of times the paper was cited.

Each of these pieces of information are given 0.5 or 1 mark depending on whether they are “partially correct” or “fully correct.” The abstracts written are free-form text with a maximum word count of 200 words. The abstract is allotted 6 marks, giving a total of 10 possible marks students can pick up.

The dataset contains 207 samples spread over 2 years using 21 different papers. The correct answers for the 4 questions listed above were extracted by selecting the most common answer in the dataset (if this system were deployed, the assessors could choose their own correct answers) as the majority of students get these questions correct.

3 The AES System

In the same way that the exercise contains two distinct types of question, these questions are scored individually by two different systems and then the scores are added to produce the final mark out of 10.

3.1 The Definite questions

The questions that have a definitive answer can be further split into two categories, the textual questions and the numerical questions.

The numerical questions are very simple to mark, the value given by the student is compared to the correct value and if the student’s value is within 10% percentage difference of the correct value the answer is deemed “fully correct” and one mark is given. If the student’s value is between 10% and 25% percentage difference of the correct valuethen the answer is deemed “partially correct” and 0.5 marks are given. If the student’s value is outside 25% then the answer is deemed “incorrect” and no marks are awarded.

One method to mark the textual questions is to extract the information from the reference e.g. Authors, Year, Journal etc. check that they are correct and in the correct order. However, in practice it was found to be extremely challenging to extract this information from the reference so this method was not used. The method used calculated the cosine similarity, as defined in[1], to compare the two texts. A cosine similarity higher than 0.9 is deemed “fully correct,” a cosine similarity between 0.65 and 0.9 is deemed “partially correct” and a cosine similarity less than 0.65 is deemed “incorrect.”

3.2 Pre-Training Dataset

Because of the small dataset, it was decided to pre-train the model on a much larger dataset. For this the datasets created for the Automated Student Assessment Prize (ASAP) sponsored by the Hewlett Foundation. For these competitions two datasets were published, one dataset for Automated Essay Scoring[2] and one for Short Answer Scoring (SAS)[3]. These datasets contain approximately 15,000 samples each, across all prompts. This allows for the AES model to begin with a much better ‘foundation’ when it begins to score abstracts. This means that the model can achieve higher performance while not overfitting to the training set. Overfitting is a particular problem as it means that the model cannot generalise from the training set, in effect ‘memorising’ each sample in the training set rather than finding patterns.

The datasets contain texts from a variety of subjects including Physics, English, Biology and Chemistry among others. This variety makes a combined dataset suitable for training a subject-agnostic AES model that can be fine-tuned on a specific question with a small dataset to achieve higher performance. Each sample was min-max normalised to a range $[0, 1]$, this was done according to the question the sample was answering, effectively converting each mark to a percentage, i.e. a fully correct answer achieves a score of 1, regardless of how many total marks are available for the question.

3.3 Model

The model used for the AES portion of the task uses a Bidirectional Long Short-Term Memory (BiLSTM)[4] encoder with an attention mechanism[5]. The input sequence is tokenised using the same tokeniser as used in the BERT transformer model[6]. This tokeniser is used because of its ability to tokenise to word parts rather than whole words. This means that when technical language is used, as is often the case with technical texts, such as abstracts, no vocabulary needs to be added, this then improves the effectiveness of pre-training such a model.

3.4 Metrics

3.4.1 Training Metric

To train the AES model a dynamic loss function was used that combines the error in the standard deviation of the model’s predictions and the Mean Squared Error (MSE) metric. Multiple loss functions can be combined using a weighted sum. This weight, p is defnied in Eq.1.

$$p(t) = \min\left(a, a \cdot \exp\left(-c\left(\frac{t}{T} - b\right)\right)\right) \tag{1}$$

where a, b and c are constants, t is the current training step and T is the total number of training steps. Such that:

$$L_T(t) = p(t) \cdot L_1 + (1 - p(t)) \cdot L_2 \tag{2}$$

where L_T is the total loss, and L_1 and L_2 are two loss functions. For this dynamic loss function L_1 is the error in the standard deviation as defined in Eq.3 and L_2 is the MSE metric as defined in Eq.4.

$$\mathrm{STDE}(\mathbf{x},\mathbf{y}) = |\sigma(\mathbf{y}) - \sigma(\mathbf{x})| \tag{3}$$

where $\mathbf{y}$ are the model's predictions, $\mathbf{x}$ are the correct values and σ is the function calculating the standard deviation.

$$\mathrm{MSE}(\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{i=0}^{N}(y_i - x_i)^2 \tag{4}$$

where N is the total number of samples.

3.4.2 Evaluation Metrics

To evaluate the performance of the AES model several metrics are used, primarily the coefficient of determination r^2 as defined in Eq.5 but also the Mean Absolute Error (MAE), defined in Eq.6 and Root Mean Squared Error (RMSE) defined in Eq.7.

$$r^2 = 1 - \frac{\sum_{i=0}^{N}(y_i - x_i)^2}{\sum_{i=0}^{N}(y_i - \bar{y})} \tag{5}$$

where $\bar{y}$ is the mean of the predicted values.

$$\mathrm{MAE}(\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{i=0}^{N}|y_i - x_i| \tag{6}$$

$$\mathrm{RMSE}(\mathbf{x},\mathbf{y}) = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(y_i - x_i)^2} \tag{7}$$

4 The AFG System

References

(1) Tashu, T. M.; Horváth, T. Semantic-Based Feedback Recommendation for Automatic Essay Evaluation. In *Intelligent Systems and Applications*; Bi, Y., Bhatia, R., Kapoor, S., Eds.; Advances in Intelligent Systems and Computing; Springer International Publishing: Cham, 2020; pp 334–346. https://doi.org/10.1007/978-3-030-29513-4_24.

(2) The Hewlett Foundation: Automated Essay Scoring.

(3) The Hewlett Foundation: Short Answer Scoring.

(4) Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **1997**, *9* (8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

(5) Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate, 2016. https://doi.org/10.48550/arXiv.1409.0473.

(6) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]* **2019**.