Automatic Grading of Online Formative Assessments using Bidirectional Recurrent Neural Networks and Attention Mechanism

Advay Pakhale, Xavier Lien, Tan Guoxian
Raffles Science Institute
Raffles Institution
Singapore, Singapore
Email: advay.pakhale@gmail.com, xavilien@gmail.com, guoxian.tan@ri.edu.sg

***Abstract*—Formative assessments have been shown to be highly beneficial for students' learning processes due to their ability to provide feedback to both teachers and students. However, the marking of short-answer questions in formative assessments is a tedious task for teachers. The advent of online learning platforms, however, has allowed for the digitalisation of student answers which opens up opportunities for automatic grading. We propose novel automatic grading architectures that (1) produce an accurate quantitative score in order to expedite the marking process and (2) provide qualitative feedback to teachers and students in terms of the key areas of improvement. These architectures consist of bidirectional long short-term memory and gated recurrent unit networks with an attention mechanism for quantitatively scoring answers, and a cosine similarity-based model that provides qualitative feedback based on a simple marking scheme comprising marking points. We evaluate these architectures across different metrics on two datasets collected from an online physics quiz, consisting of two short-answer physics questions on the topic of thermodynamics. We show that our architectures achieve reasonable accuracy on the scoring task and provide useful feedback to teachers and students, thus successfully aiding in automatically grading formative assessments.**

***Keywords*—automatic grading; machine learning; online learning; bidirectional recurrent neural networks; attention mechanism; formative assessments**

I. Introduction

Formative assessments place a primary focus on providing qualitative feedback for both teachers and students. Such feedback not only allows teachers to monitor their students' progress and evaluate the effectiveness of their own instruction, but also provides insight as to how the teaching and learning process can be optimised going forward [1]. Formative assessments have been shown to produce a much greater impact on student learning as compared to summative assessments when administered frequently and in a timely manner [2]. Hence, formative assessments clearly play a beneficial role in today's modern classrooms.

However, grading certain forms of formative assessments such as short-answer questions is a complex task that requires significant human intervention and input. Furthermore, valuable feedback needs to be produced that is beneficial to both students and teachers, which is a time-consuming and tedious task for teachers. The automation of this task would allow open-ended formative assessments to be administered more frequently and easily, greatly easing the workload of teachers and allowing both students and teachers to reap their benefits through the frequent and timely report of progress that they would receive.

The advent of online learning platforms has made administering formative assessments more convenient and more importantly, has allowed for the digitalisation of student answers which opens up opportunities for automatic grading that are explored in the following sections.

II. Literature Review

Much progress has been made in developing different approaches to the automatic grading of open-ended questions. Reference [3] showed that a system utilising multiple linear regression with a combination of hand-crafted features and probabilistic models, such as Bayesian classifiers and k-nearest-neighbour algorithms, can give results comparable with human graders. Reference [4] developed CarmelTC, a rule-based approach, combining both features obtained from deep syntactic functional analyses of texts and a "bag-of-words" classification extracted from Rainbow Naive Bayes, outperforming Latent Semantic Analysis, Rainbow Naive Bayes and a purely symbolic approach. Reference [5] used a deep learning approach for essay grading while [6] used handpicked features with linear regression on the same dataset. The former performed significantly better, demonstrating the superiority of deep learning approaches over classic Natural Language Processing (NLP) methods.

However, much of the literature focuses on grading summative assessments; the only objective is to produce an accurate quantitative prediction of the score. This is arguably easier and less complex than grading formative assessments which involves producing feedback for both teachers and students on top of merely providing a quantitative score. Thus, there is a need for new automatic grading architectures that can fulfil the aims of formative assessments.

III. Methodology

A. Datasets

An online physics quiz was conducted on 292 Raffles Institution Secondary 2 students, comprising two short-answer physics question on the topic of thermodynamics. Thermodynamics was chosen as it is an important topic in mainstream schools which students have many misconceptions about, and this topic is typically tested using qualitative short-answer questions. The first question is a simple recall question, while the second question is a more complex application question, designed to have more variance in answers. The answers were then graded by an entire level of 3 physics teachers based on a marking scheme comprising different marking points. Student answers were then pre-processed by tokenising and converting to lowercase while punctuation, non-alphabetical characters and stop words were also removed. The answers and scores from question 1 and question 2 are denoted dataset 1 and dataset 2 respectively.

B. Components of Architecture

We aim to design an architecture (Fig. 1) that automatically grades short-answer formative assessments by **(1)** providing an accurate quantitative score for student answers in order to expedite the marking process and **(2)** providing qualitative feedback to teachers and students in terms of the key areas of improvement. We evaluated different qualitative and quantitative models that attempt to achieve these two aims.

1) Quantitative Component: We propose a neural network model for the quantitative component. In order to design the best performing architecture, we propose different types of models below. Firstly, we compare the use of different word embeddings. Secondly, we fix our baseline model to be a simple feedforward neural network. Thirdly, we compare two types of Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks and Gated Recurrent Unit (GRU) networks. Fourthly, we compare the use of bidirectional RNNs (BiRNNs) and their unidirectional variants. Finally, we propose an attention mechanism.

a) Word Embeddings: Within the neural networks, the answers are first represented in terms of word vectors that can encode meaningful semantic relationships between words. We compare 2 types of word embeddings - GloVe [7] and fastText [8], both of which are 300 dimensional.

b) Feedforward Neural Network (Baseline): We chose a simple feedforward neural network as our baseline. This network is trained on an ”answer vector” for each answer, which is simply the sum of the word vectors for each word in the answer. While this approach does not preserve sequential information, word vectors have been shown to be able to meaningfully encode linearities such as *king - man + woman = queen* and *Paris - France + Italy = Rome* [9]. Thus, we hypothesise that the answer vectors will nevertheless be able to encode meaningful information about the answers to a certain degree. The architecture of the baseline and non-baseline models can be found in Fig. 2.

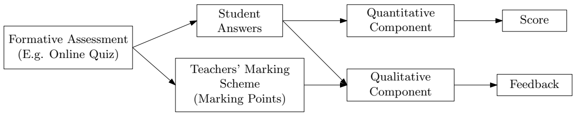
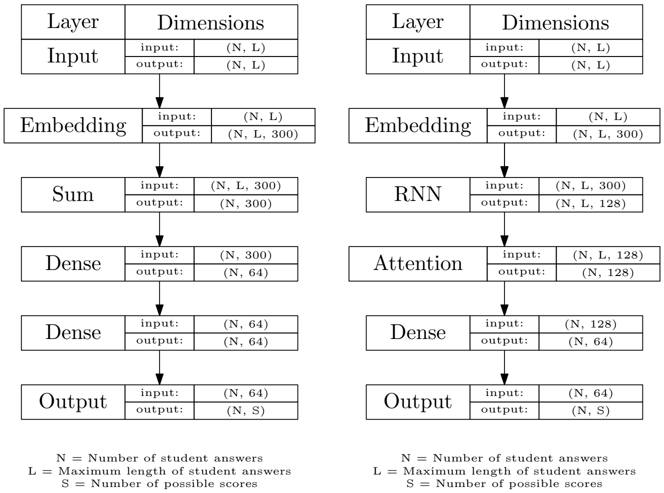


Fig. 1. Outline of Architecture

c) LSTM and GRU Networks: LSTM networks [10] preserve long-distance dependencies, making them ideal for processing text sequences [11]. On the other hand, GRU networks were developed as an alternative to LSTM networks and are computationally less complex [12], yet perform comparably to LSTM networks in sequence processing tasks [13]. We chose to work with RNNs as they have demonstrated excellent performance in sequence processing tasks, including sequence classification, due to their ability to account for sequential information [14], making them applicable to our task of grading short-answer physics questions.

d) Bidirectional Recurrent Neural Networks: BiRNNs [15] such as BiLSTM networks aim to improve upon RNNs by processing the same input sequence twice, forwards and backwards. This allows the BiRNN to retain contextual information in both directions and has led to an improvement in performance in various sequence processing tasks [16], [17]. Hence, we compare the performance of BiLSTM and BiGRU models with that of regular LSTM and GRU models.

e) Attention Mechanism: Recently, the attention mechanism has been developed for sequence processing tasks such as machine translation and sequence classification [18], [19]. Since not all words in a sentence contribute equally to its meaning, attention is used to place more weight on more important words while placing less weight on less important words. We hypothesise that attention would be able to extract keywords from answers to aid in marking them more effec-



(a) Baseline

(b) Non-Baseline

Fig. 2. Model Architectures

tively, similar to how a teacher might mark answers. A detailed diagram showing the architecture of the attention layer on top of a BiRNN layer can be found in Fig. 3.

2) Qualitative Component: We propose a few different methods to provide qualitative feedback to students and teachers using the marking schemes in our datasets by capturing common strengths and weaknesses in student answers. We mainly adapt and draw inspiration from classic probabilistic topic models such as Latent Dirichlet Allocation [20], which produces sparse and low-dimensional interpretations of topic memberships in documents. These are highly interpretable, allowing humans to gain high-level insight and intuition from them, which is especially necessary in the field of formative assessment where human understanding and feedback is key.

a) Vector Decomposition: Our datasets provide us with marking points and student answers. In order to manipulate these mathematically, a vector representation first needs to be defined for them. Motivated by the fact that word vectors can meaningfully encode linearities, as mentioned in Section III-B1b, we initialise the marking point vectors to be the sum of the words in each marking point, so that they capture the meaning of the marking points:

$$\mathbf{t_k} = \sum_{i \in T_k} \mathbf{w_i}\,, \tag{1}$$

where $\mathbf{t_k}$ and T_k are the marking point vector and the set of words in the kth marking point out of n marking points respectively, and $\mathbf{w_i}$ is the word vector for the ith word. The GloVe word vectors are used for the purposes of the qualitative experiments.

A similarly suitable vector representation of student answers, is simply the sum of all the word vectors for each word in the answer:

$$\mathbf{a_j} = \sum_{i \in A_j} \mathbf{w_i}\,, \tag{2}$$

where $\mathbf{a_j}$ and A_j are the answer vector and the set of words in the jth answer respectively. However, an answer can also be represented as a complete or incomplete combination of the marking points.

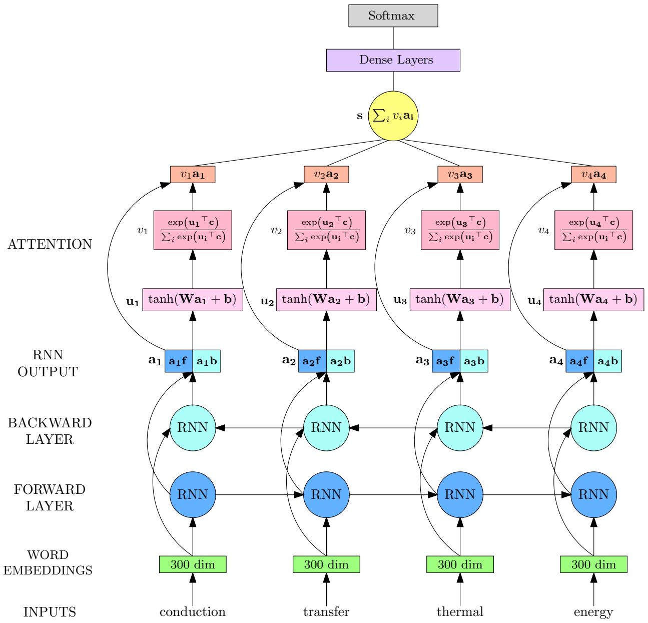

Softmax
Dense Layers
s
$\sum_i v_i \mathbf{a_i}$
$v_1\mathbf{a_1}$
$v_2\mathbf{a_2}$
$v_3\mathbf{a_3}$
$v_4\mathbf{a_4}$
ATTENTION
v_1
$\frac{\exp(\mathbf{u_1}^\top \mathbf{c})}{\sum_i \exp(\mathbf{u_i}^\top \mathbf{c})}$
v_2
$\frac{\exp(\mathbf{u_2}^\top \mathbf{c})}{\sum_i \exp(\mathbf{u_i}^\top \mathbf{c})}$
v_3
$\frac{\exp(\mathbf{u_3}^\top \mathbf{c})}{\sum_i \exp(\mathbf{u_i}^\top \mathbf{c})}$
v_4
$\frac{\exp(\mathbf{u_4}^\top \mathbf{c})}{\sum_i \exp(\mathbf{u_i}^\top \mathbf{c})}$
$\mathbf{u_1}$ $\tanh(\mathbf{W}\mathbf{a_1}+\mathbf{b})$
$\mathbf{u_2}$ $\tanh(\mathbf{W}\mathbf{a_2}+\mathbf{b})$
$\mathbf{u_3}$ $\tanh(\mathbf{W}\mathbf{a_3}+\mathbf{b})$
$\mathbf{u_4}$ $\tanh(\mathbf{W}\mathbf{a_4}+\mathbf{b})$
RNN OUTPUT
$\mathbf{a_1}$ $\mathbf{a_1 f}$ $\mathbf{a_1 b}$
$\mathbf{a_2}$ $\mathbf{a_2 f}$ $\mathbf{a_2 b}$
$\mathbf{a_3}$ $\mathbf{a_3 f}$ $\mathbf{a_3 b}$
$\mathbf{a_4}$ $\mathbf{a_4 f}$ $\mathbf{a_4 b}$
BACKWARD LAYER
RNN
RNN
RNN
RNN
FORWARD LAYER
RNN
RNN
RNN
RNN
WORD EMBEDDINGS
300 dim
300 dim
300 dim
300 dim
INPUTS
conduction
transfer
thermal
energy

Fig. 3. Attention Mechanism on Top of a BiRNN

Thus, we hypothesise that the answer vector $\mathbf{a_j}$ can be decomposed into a linear combination of marking point vectors:

$$\hat{\mathbf{a_j}} = \sum_{k=1}^{n} p_{jk}\mathbf{t_k} = \sum_{k=1}^{n} p_{jk} \sum_{i \in T_k} \mathbf{w_i}\,, \tag{3}$$

where p_{jk} is the proportion of a marking point k in answer j and $p_{jk} \in [0, 1]$. $p_{jk} = 0$ would mean that a marking point is completely not within the student's answer, while $p_{jk} = 1$ would mean otherwise. Let $\mathbf{T}$ be the matrix whose columns are $\mathbf{t_1}, \ldots, \mathbf{t_k}, \ldots, \mathbf{t_n}$. Then $\hat{\mathbf{a_j}}$ is an element of the column space of $\mathbf{T}$, and can be equivalently expressed as $\hat{\mathbf{a_j}} = \mathbf{T}\mathbf{p_j}$. In order to then find an accurate decomposition of the answer vector, the optimal proportion vector $\hat{\mathbf{p_j}}$ can be found by optimising the mean squared error between $\hat{\mathbf{a_j}}$ and $\mathbf{a_j}$:

$$\hat{\mathbf{p_j}} = \arg\min_{\mathbf{p_j}} \left\|\hat{\mathbf{a_j}} - \mathbf{a_j}\right\|^2 \tag{4a}$$

$$= \arg\min_{\mathbf{p_j}} \left\|\mathbf{T}\sigma(\mathbf{p_j}) - \mathbf{a_j}\right\|^2 . \tag{4b}$$

The logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$ is applied element-wise on $\mathbf{p_j}$ to enforce the constraint $0 \le p_{jk} \le 1$.

b) Cosine Similarity: Under this method, the marking point proportions p_{jk} are defined as the cosine similarity [21] between the student answer vector and each marking point vector:

$$p_{jk} = \cos\theta_{jk} = \frac{\mathbf{a_j} \cdot \mathbf{t_k}}{\|\mathbf{a_j}\|\|\mathbf{t_k}\|}\,. \tag{5}$$

These p_{jk} values are collected into a vector, $\mathbf{p_j} = \begin{pmatrix} p_{j0} & p_{j1} & \cdots & p_{jk} \end{pmatrix}$, for each answer j.

c) Euclidean Distance: p_{jk} can also defined as the Euclidean distance [21] between the student answer vector and each marking point vector:

$$p_{jk} = \|\mathbf{a_j} - \mathbf{t_k}\|\,. \tag{6}$$

Similarly, we define $\mathbf{p_j} = \begin{pmatrix} p_{j0} & p_{j1} & \cdots & p_{jk} \end{pmatrix}$.

IV. Data and Discussion

A. Evaluation Methodology

The datasets are split into training and validation sets, with a proportion of 0.8 and 0.2 respectively. Each model is evaluated using 5-fold cross-validation on 3 metrics. Firstly, accuracy checks to see whether the model is able to give each student answer a correct score. Such a metric is useful for teachers and students as an ideal model would be able to mark all student answers exactly like a teacher would. Secondly, categorical cross-entropy loss is used to compare the different models as this the loss function that they were trained using.

Lastly, weighted F1 score takes into consideration the unequal distribution of scores for the student answers as there are very few students who received full marks for each question. The hyperparameters used for the quantitative models can be found in Table I.

B. Evaluation of Quantitative Component

1) Comparison of Models:

a) LSTM vs GRU: From Tables II and III, the GRU models performed better than the LSTM models on dataset 1, while the opposite is true for dataset 2. We hypothesise that this is the case because of the difference in complexity of the two questions. Since question 1 is less complex than question 2, the simpler GRU models with fewer parameters are likely to have less overfitting than the LSTM models. Likewise, the LSTM models are likely able to better capture the complexities of the second questions, giving them better performance. As shown by [22], LSTM networks are ”strictly stronger” than GRU networks as they can easily perform unbounded counting, which GRU networks cannot, further supporting the hypothesis that LSTM networks are better suited for more complex questions. The difference in question complexity is likely also the reason for the disparity in performances between the two datasets for all models.

TABLE I
HYPERPARAMETERS FOR QUANTITATIVE MODELS

GloVe Dimensions	300d
Number of Hidden Layers (without attention)	2
Number of Hidden Layers (with attention)	3
Number of Units per Hidden Layer	64
Activation	ReLu
Optimizer	Adam
Learning Rate	0.001
Dropout	0.1
Loss	Categorical Cross-Entropy

TABLE II
PERFORMANCE OF QUANTITATIVE MODELS ON DATASET 1

Models	Accuracy	Loss	F1 Score
Feedforward Neural Network (Baseline)	0.733	0.849	0.719
LSTM with GloVe	0.702	0.780	0.660
BiLSTM with GloVe	0.705	0.837	0.699
BiLSTM with GloVe and Attention	0.760	0.698	0.750
BiLSTM with fastText and Attention	0.716	0.741	0.696
GRU with GloVe	0.715	1.050	0.701
BiGRU with GloVe	0.709	0.909	0.692
BiGRU with GloVe and Attention	**0.781**	**0.701**	**0.775**
BiGRU with fastText and attention	0.726	0.715	0.692

TABLE III
PERFORMANCE OF QUANTITATIVE MODELS ON DATASET 2

Models	Accuracy	Loss	F1 Score
Feedforward Neural Network (Baseline)	0.472	1.354	0.433
LSTM with GloVe	0.428	1.840	0.399
BiLSTM with GloVe	0.432	1.546	0.394
BiLSTM with GloVe and Attention	0.520	1.530	0.500
BiLSTM with fastText and Attention	**0.541**	**1.157**	**0.506**
GRU with GloVe	0.414	1.658	0.373
BiGRU with GloVe	0.421	1.459	0.392
BiGRU with GloVe and Attention	0.462	1.435	0.447
BiGRU with fastText and attention	0.496	1.265	0.465

b) Attention: In addition, attention significantly improved the performance of the models, supporting our initial hypothesis. This is likely because physics answers are marked based on keywords, which is suited for attention as it is able to place weight on more important words and extract keywords. This is further corroborated by the attention weights extracted from the layer, visualised in Fig. 4, which shows that attention places more weight on keywords such as “gases”, “solids”, and “states”.

c) Word Embeddings: GloVe performed better for the first dataset while fastText performed better for the second dataset. We hypothesise that this performance disparity could be to the fact that fastText was better able to capture the more complex nature of the second question, due to the ”subword” information captured by fastText [8].

d) Baseline: Contrary to our initial hypothesis, the baseline performed surprisingly well and outperformed many of the more complex models, despite having a major disadvantage due to the fact that it does not preserve sequential information. We conjecture that this could be due to two possible reasons. The more complex models have a significantly greater number of parameters compared to the baseline. This, combined with the relatively small datasets, could have led to overfitting in these models [23], preventing them from generalising their classification task to the validation set as well as the baseline model, leading to the latter having a better performance. The baseline could have also outperformed the other models due to poor hyperparameter tuning in the more complex models, causing them to be stuck in poor local minima [24]. If this is the case, then there is a possibility that performance can be significantly improved with a more careful hyperparameter search.

2) Performance Across Different Data Environments: We explored how the number of training samples affects the

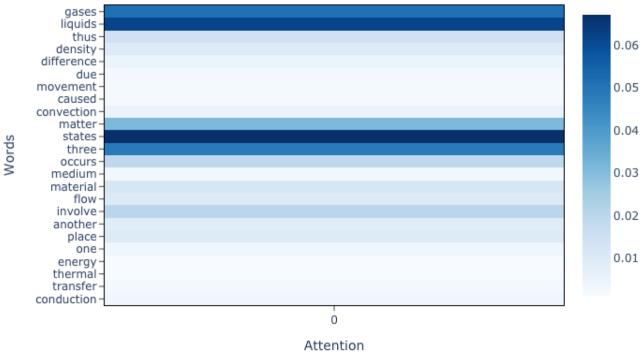


Fig. 4. Attention Weights of Answer Sample from Dataset 1

models' performance. 42 student answers were put aside to be the validation set. The size of the training set was varied from 25 samples to 250 samples in increments of 25 samples and the performance of the model was monitored. The detailed results, which can be found in Figs. 5 and 6, show that our best models perform well even in low data environments. Since one of the intended aims of our models is to reduce the marking load for teachers, this demonstrates their utility as teachers only need to mark a small proportion of the dataset for the models to perform well.

C. *Evaluation of Qualitative Component*

1) Comparison of Models: To test how accurate the set of $\mathbf{p_j}$ vectors generated by each qualitative model is, we trained a simple feedforward neural network to predict the scores based on $\mathbf{p_j}$ as the input. This is based on the assumption that accurate proportions of each marking point should be correlated to the score of the answers, since these marking points are used by teachers to mark the answers. The results are summarised in Tables IV and V. It can be seen that the cosine similarity model clearly outperforms the rest and the results achieved by it are even comparable to our best quantitative models despite the relative simplicity of this approach. This could be due to the same reasons our baseline quantitative model outperformed more complex models, as highlighted in Section IV-B1d. This good performance also shows that the proportions generated are accurate.

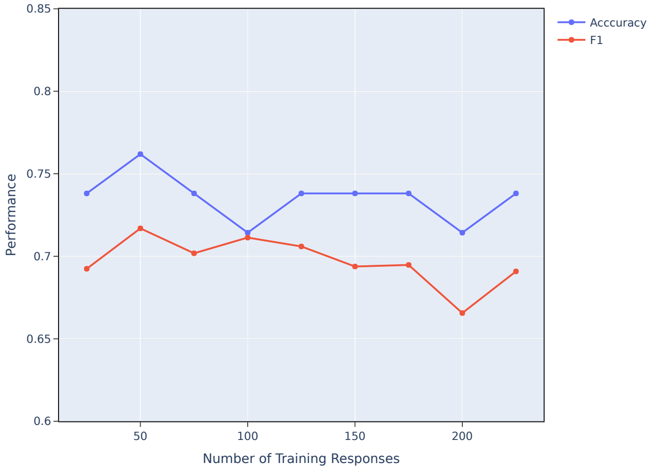


Fig. 5. Performance of Dataset 1 Best Model (BiGRU with GloVe and Attention) Against Size of Training Set

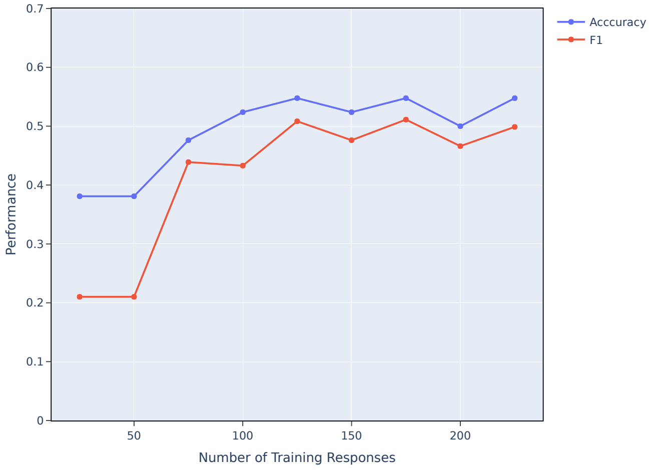


Fig. 6. Performance of Dataset 2 Best Model (BiLSTM with fastText and Attention) Against Size of Training Set

2) Producing Feedback: The $\mathbf{p_j}$ vectors can be used to produce feedback for both teachers and students. Upon some basic visualisation, students can view their own $\mathbf{p_j}$ vector and observe which marking points they managed to incorporate into their answer and which marking points were missed out, to give them an indication of their areas for improvement. Furthermore, an average $\mathbf{p_j}$ vector across all answers can also be calculated and visualised for teachers' analysis, as seen in Fig. 7. Fig. 7 was produced by averaging the $\mathbf{p_j}$ vectors from the cosine similarity model run on dataset 1. From this, teachers can immediately observe which marking points most students wrote in their answers and which they failed to incorporate. This would provide feedback to teachers about which parts of their teaching were well understood by the students and likewise, which parts might need review. We can thus see how the qualitative feedback produced by the model aids in the task of formative assessment.

TABLE IV
PERFORMANCE OF QUALITATIVE MODELS ON DATASET 1

Models	Accuracy	Loss	F1 Score
Vector Decomposition	0.591	0.931	0.552
Cosine Similarity	**0.716**	**0.652**	**0.713**
Euclidean Distance	0.648	0.769	0.608

TABLE V
PERFORMANCE OF QUALITATIVE MODELS ON DATASET 2

Models	Accuracy	Loss	F1 Score
Vector Decomposition	0.455	1.251	0.438
Cosine Similarity	**0.541**	**1.128**	**0.527**
Euclidean Distance	0.534	1.165	0.493

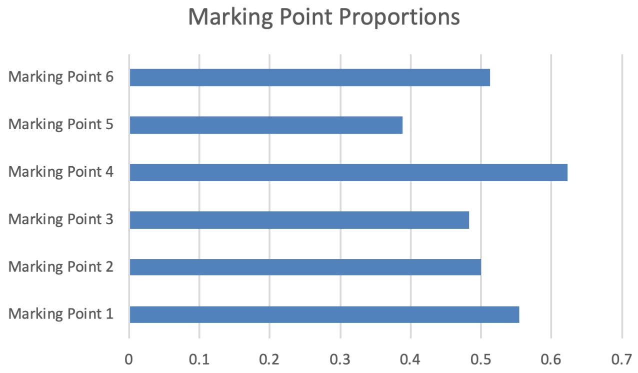


Fig. 7. Marking Point Proportions

D. Final Architecture

Based on our results, we recommend two separate architectures for different types of questions. For the quantitative component, we recommend a BiGRU network with GloVe embeddings and an attention mechanism for simple recall style questions but a BiLSTM network with fastText embeddings and an attention mechanism for more complex application-based questions. For the qualitative component, we recommend the cosine similarity model for both types of questions.

V. Conclusion and Future Work

In this work, we proposed a novel approach to grading short-answer physics questions. Both the qualitative and quantitative components are shown to perform well, especially in low data conditions, which is important in order to reduce the marking workload of teachers. Furthermore, the architectures also provide interpretable feedback for both teachers and students, aiding in the task of formative assessment.

Given such promising results, these architectures can be applied to online learning portals where teachers can deploy a system for each short-answer question they create and feed the system a small amount of marked answers as training data along with a simple marking point-based marking scheme.

In future works, we propose exploring different word embeddings such as ELMo [25], which could yield a performance upgrade. Additionally, to tackle the problem of overfitting we identified, dropout layers [26] could be employed in our models. Thirdly, more careful hyperparameter optimisation using comprehensive methods such as randomised search [27] and sequential search [28] on our current models could yield better results. Considering the good performance of the qualitative models, it may be possible to increase performance by incorporating the marking points as features to the quantitative models. Lastly, the models could also be evaluated on other datasets, such as larger datasets and datasets from different domains such as other sciences, to test its scalability and adaptability.

References

[1] P. Black and D. Wiliam, "Developing the theory of formative assessment," *Educational Assessment, Evaluation and Accountability (formerly: Journal of Personnel Evaluation in Education)*, vol. 21, no. 1, p. 5, 2009.

[2] ——, "Assessment and classroom learning," *Assessment in Education: Principles, Policy & Practice*, vol. 5, no. 1, pp. 7–74, 1998.

[3] L. S. Larkey, "Automatic essay grading using text categorization techniques," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '98. New York, NY, USA: ACM, 1998, pp. 90–95.

[4] C. P. Rosé, A. Roque, D. Bhembe, and K. VanLehn, "A hybrid approach to content analysis for automatic essay grading," in *Companion Volume of the Proceedings of HLT-NAACL 2003 - Short Papers*, 2003, pp. 88–90. [Online]. Available: https://www.aclweb.org/anthology/N03-2030

[5] H. Nguyen and L. Dery, "Neural networks for automated essay grading," 2018.

[6] Y. Zhang, R. Shah, and M. Chi, "Deep learning+ student modeling+ clustering: A recipe for effective automatic short answer grading." *International Educational Data Mining Society*, 2016.

[7] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[8] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] ——, "Lstm can solve hard long time lag problems," in *Proceedings of the 9th International Conference on Neural Information Processing Systems*, ser. NIPS'96. Cambridge, MA, USA: MIT Press, 1996, pp. 473–479.

[12] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014.

[13] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014.

[14] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," 2016.

[15] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[16] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional lstm networks for improved phoneme classification and recognition," in *International Conference on Artificial Neural Networks*. Springer, 2005, pp. 799–804.

[17] M. Sundermeyer, T. Alkhouli, J. Wuebker, and H. Ney, "Translation modeling with bidirectional recurrent neural networks," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 14–25.

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014.

[19] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 1480–1489. [Online]. Available: https://www.aclweb.org/anthology/N16-1174

[20] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944937

[21] A. Huang, "Similarity measures for text document clustering," in *Proceedings of the 6th New Zealand Computer Science Research Student Conference*, vol. 4, 01 2008, pp. 9–56.

[22] G. Weiss, Y. Goldberg, and E. Yahav, "On the practical computational power of finite precision rnns for language recognition," 2018.

[23] D. M. Hawkins, "The problem of overfitting," *Journal of Chemical Information and Computer Sciences*, vol. 44, no. 1, pp. 1–12, 01 2004.

[24] N. Reimers and I. Gurevych, "Optimal hyperparameters for deep lstm-networks for sequence labeling tasks," 2017.

[25] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[27] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[28] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 2546–2554. [Online]. Available: http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf