Exploring Pre-scoring Clustering for Short Answer Grading

L. Petricioli, K. Skračić, J. Petrović and P. Pale
University of Zagreb Faculty of Electrical Engineering and Computing, Zagreb, Croatia
{Lucija.Petricioli, Kristian.Skracic, Juraj.Petrovic, Predrag.Pale}@fer.hr

***Abstract* – Automatic short answer grading is a topic that has gained significant popularity recently, especially due to developments in natural language processing. While automated grading in computer supported assessment tasks traditionally imposed significant restrictions on the answer format (e.g., multiple choice questions), automated short answer grading could enable assessment scalability with very few answer format limitations and thereby increase the assessment tasks' validity. Here, 'short answer' refers to a text of up to, approximately, 10 sentences. However, automatic solutions require a lot of pre-graded material. In this paper, several pre-trained machine learning models were utilized to explore pre-scoring clustering for short answer grading of text in Croatian. The aim of this approach is to shorten the process of manual short answer grading by clustering similar answers, facilitating the development of automatic grading solutions. The described approach was evaluated on a dataset containing graduate students' answers in Croatian to six questions related to cyber security topics. The obtained results are promising and show how increases in cluster purity, normalized mutual information, Rand index, and adjusted Rand index measures can be achieved by finetuning a pre-trained model.**

Keywords – automatic short answer grading; ASAG; semi-automated short answer scoring; short answer grading; short text; short answer; automatic grading; natural language processing

I. INTRODUCTION

The ability to assess the knowledge and skills of a large number of students in a remote, quick, and scalable way has been a topic of interest for quite some time [1]–[3], and it has gained additional attention since the outbreak of the COVID-19 pandemic, when a lot of formal education had to be abruptly shifted into online environments. Online assessments are already used in massive open online courses, various paid platforms, distance learning, hybrid courses that are a part of formal education, and many others. The importance of assessment in learning can hardly be overestimated since it can be used to assist learning through not only getting students to perform knowledge retrieval [4] or through enabling timely feedback on assessment outcomes [5] (formative assessment) but also through evaluating the students' achieved learning outcomes for course completion or certification (summative assessment).

Automatic assessment grading has been a topic of interest for decades already. Whether an assessment item can be automatically graded, however, depends on the assessment item type. Items with more constrained answer types [6], like multiple choice questions, are easy to grade automatically. However, their ecological validity and ability to assess higher learning outcomes are, at times, debated – some authors point out that multiple choice questions do not support the same kind of knowledge retention and synthesis as, for example, short answer questions do [7].

Short answer questions are questions that require a short response written in one's own words [8]. Answering a short answer question can, therefore, require understanding, whereas answering multiple-choice questions can be based on solely recognizing the correct answer [1], [2]. Unsurprisingly though, this question type requires much more effort to grade, not only manually but also automatically. Automatic Short Answer Grading (ASAG) is based on Natural Language Processing (NLP) and is certainly a difficult problem to solve. Even automation does not absolve the human graders of their task – namely, all automatic grading solutions require pre-graded content to extract features from. Machine Learning (ML) solutions are especially notorious for requiring large amounts of data [8]–[10].

Since manually scoring answers to short answer questions is time-consuming and exhausting, and fully automating short answer scoring a difficult NLP task that cannot really be performed in absence of any manual scoring, a possible reasonable trade-off could be found in semi-automation [1], [8]–[15]. Semi-automated short answer scoring techniques aim to reduce the time human graders need to grade short answers, either for directly grading exams or for annotation needed for further automatization [11].

This paper explores several pre-trained machine learning models and how finetuning one of them improves the results of pre-scoring clustering for grading short answers in Croatian. The goal of this is to reduce the time needed to score short text answers manually and the approach taken provides promising results. This paper is organized as follows: in section II, related semi-automated approaches to short answer grading are presented; section III describes the explored approach to pre-scoring short answers, the dataset and evaluation procedures used; section IV showcases the results, and section V presents the authors' conclusions.

II. Related work

Many of the approaches dealing with short answer scoring semi-automation are based on the same premise: lexically similar answers should likely be scored similarly [8], [11], [13]. The data sets used in each of the following methods are pre-graded to enable testing the various approaches.

Basu et al. [1] propose an approach in which they cluster answers based on a similarity metric, where each cluster is subdivided into subclusters, resulting in a cluster hierarchy. The authors' idea is that a grader can grade a cluster as a whole, thereby significantly reducing the time needed for manual grading. The subclusters are there to help with a finer grading granulation – if a subcluster does not really fulfil the criteria for the score of the cluster, the subcluster can be assigned a different grade easily. Furthermore, clustering answers could help teachers determine whether their students have a common misconception regarding a question, which can help them rectify it quickly [1], [8].

Horbach et al. [11] experiment with clustering answers, grading one answer per cluster and then propagating the grade throughout the cluster. According to their analyses, grading the answer closest to the centroid of a cluster and propagating the grade to the whole cluster yields good results. They can achieve an accuracy of 85% by manually grading just 40% of the answer data set.

Wolska et al. [15] take a slightly different approach to answer clustering – their clustering is aimed at speeding up answer grading by showing the graders similar answers one after the other. Their theory is that reducing "context switching" (jumping from grading correct answers to incorrect ones and back) reduces the time needed for grading all the student answers. Ultimately, they get mixed results. Pado et al. [13] present a very similar idea to the one in [15] and, consequently, get similar results.

Zesch et al. [14] are mostly interested in quicker annotation of training examples for automatic short answer scoring solutions based on ML. They test several clustering solutions and conclude clustering is best suited for very short answers (phrases of up to three words) that are subsequently easy to separate from other phrases. Longer answers can have a lot of lexical overlap but have completely different meaning, making them difficult to cluster in a meaningful manner.

Horbach et al. [9] switch to Active Learning (AL) as their method of choice. They adopt an iterative approach: the teacher is presented with a number of answers to grade. The resulting graded answers are used to train a sequential minimal optimization classifier which is then run on the test data set (pre-graded), and the results of this experiment are then evaluated. This sequence of steps can be repeated until the whole data set is graded or up to a certain threshold. The authors find this AL approach works well for *some* questions; they especially note that questions that have (at least some) clearly separable answer classes are good candidates.

Horbach et al. [8] return to clustering once again. This time they propose manual grading before and during clustering to achieve better results. Namely, the teacher needs to grade some answers before anything is clustered. The graded answers are then used for feature extraction – the authors argue this can reduce the noise in the features, thereby improving clustering results. They also "reuse" the graded answers during clustering in two ways: firstly, they use them as seeds for the clustering algorithm, and, secondly, they construct relational constraints between them (e.g., whether two answers should or should not be part of the same cluster). Finally, they continue with the centroid grade propagation approach from [11]. They show that their multifaceted approach yields promising results.

Mieskes et al. [12] propose an ensemble of automatic graders as the base for their solution. They automate the grading process with three graders; one based on random forests, one on decision trees, and one on support vector machines. Manual grading is employed in those cases where the automatic graders disagree. With this approach, the authors achieve a reduction of effort needed for manual grading of up to 75% in the case of answers with a binary class (correct/incorrect), and up to 40% for those answers that have a more complex grading scheme.

On the other hand, Tashu et al. [10] opt for locality sensitive hashing as a way of both speeding up and improving answer clustering attempts. They start with a small subset of graded answers that can be considered seeds for clustering. They then hash each new answer and calculate its distance to currently graded answers – the new answer is then assigned the grade of the answer it is closest to. If there are multiple graded answers at the same minimal distance to the new answer, their grades are averaged, and the resulting score is assigned to the new answer.

Generally, many of the authors test their approaches on data sets commonly used for Automatic Short Answer Grading (ASAG). Two of the most used data sets are the PowerGrading (PG) [1], [2], [8], [14] and the Automated Student Assessment Prize (ASAP) [2], [8], [10], [12], [14], [16] data set. However, some of the researchers note that the PG data set is lacking – it has very short answers that often consist of a single phrase and resemble fill-the-gap exercises [8], [14], making them perfectly suitable for clustering approaches, but providing little value to any purported innovative approaches tested solely on it. On the other hand, the ASAP data set has proved to be a clustering challenge with its high number of tokens (median around 48 and maximum of up to 66 per answer [2]) and considerable lexical variance.

III. Pre-scoring clustering approach

This section describes the data set used for the research presented in this paper, as well as the experimental setup and the explored approach.

A. Data set

The data set used in this research consists of six questions in the field of Cyber Security (CS) and their corresponding answers. The answers were provided by graduate students studying either electrical engineering or computing. There are 72 answers per question with an average number of tokens of around 18. All the answers were graded by a single grader and scored with 0-2 points or 0-3 points, depending on the question. The language of

both the questions and the answers is Croatian. The questions the students had to answer were the following (translated to English; the point range for each question is listed in brackets next to it):

Q1: What does a hacker do? (0-3)

Q2: What is cryptography? (0-2)

Q3: What is phishing? (0-3)

Q4: What is social engineering? (0-3)

Q5: What is the difference between phishing and spam (in the context of e-mail)? (0-3)

Q6: How would you check the credibility of information found on the Internet? (0-2)

It is important to note that the choice of language does limit the number of applicable approaches.

B. Experimental setup

In order to enable the calculation of distances between answers and, therefore, their subsequent clustering, their text needs to be vectorized (as naïve approaches like Levenshtein distance would not yield usable results on text of up to or approximately 10 sentences in length [17]). Three standard text vectorization approaches were used for benchmarking:

- **FastText**[1], an open-source library often used for learning text representations and classifiers. It offers pre-trained word vectors for 157 languages, one of which is Croatian. These vectors were used as a baseline in this experiment, since many existing approaches use static word vectors and n-grams [2], [8], [9], [11], [12], [14], [15].
- Two models from TensorFlowHub – a 12-layer model [2] (**XLM-R-12**) and a 24-layer model [3] (**XLM-R-24**). Both models were pre-trained on the Croatian language. These two models are based on Bidirectional Encoder Representations from Transformers (BERT) [18], specifically the RoBERTa architecture [19] used for cross-lingual representation learning [20].

In addition to those three models, the described XLM-R-12 was modified to attempt to achieve better results. This model was chosen for two reasons: the dataset available is relatively small, so using a larger model would most likely not yield better results, and, more importantly, a larger model would require significantly more resources for finetuning than the researchers had available.

Changes to the standard XLM-R-12 model included adding two more layers: a fully connected layer of 512 neurons with ReLU activation and 10% dropout to prevent overfitting, and a softmax output that predicts the grade of the answer using a one-hot encoding vector. Both were added to the pooled [CLS] token output. Five random answers per question were selected from the pre-graded data set for finetuning. Once the process was complete, the softmax output layer was stripped from the model, leaving the fully connected 512-neuron layer for vectorizing the answers.

Once all the answers were vectorized with the different methods, an elbow analysis was performed to work out how many clusters would yield the best distribution of the data. The analysis showed that six clusters would be appropriate.

Finally, the vectors were all clustered using scikit-learn's [21] spectral clustering implementation[4], with the number of clusters parameter set to six and the affinity parameter set to nearest neighbours clustering. K-means and mean shift were also explored as alternatives to spectral clustering, but with weaker results. In spectral clustering, data points are regarded as nodes of a connected graph and clusters are found by partitioning this graph based on its spectral decomposition. Word embeddings are compared using the cosine distance measure because the cosine distance can measure the semantic similarity of the words. By using the pooled embedding of the answer, the answers are effectively grouped based on their semantic closeness.

Once the various methods' vectors were clustered, the results were evaluated using several cluster quality measures. The comparisons between two clusterings are done between the graded data set and the clustered data sets. The grade that is given to an answer in the data set is seen as the "cluster" the answer is assigned to (albeit manually). The cluster quality measures used were:

- **Cluster purity.** Cluster purity measures how often the same class is part of the same cluster. In the case of this paper, it measures how often answers that have been manually given the same grade are part of the same cluster. The higher the percentage of answers of the same grade in a cluster, the purer the cluster.
- **Normalized Mutual Information.** Normalized Mutual Information [5] (NMI) is a normalized measure of how dependent two variables, or, in this case, clusters are. Logically, if two clusters coincide perfectly, they are completely dependent, and their NMI score is 1. If they are independent, their score is 0.
- **The Rand Index.** The Rand Index[6] (RI) measures the agreement between two clusterings. For each pair of elements in both clusterings, it calculates whether the two clusterings agree on whether the elements are part of the same cluster or a different cluster. If all the pairs are distributed across clusters in the same way in both clusterings, the

[1] https://fasttext.cc/
[2] https://tfhub.dev/jeongukjae/xlm_roberta_multi_cased_L-12_H-768_A-12/1
[3] https://tfhub.dev/jeongukjae/xlm_roberta_multi_cased_L-24_H-1024_A-16/1
[4] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html
[5] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html
[6] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.rand_score.html#sklearn.metrics.rand_score

index equals 1. If the clusterings always disagree, the index is 0.

- **The Adjusted Rand Index.** The Adjusted Rand Index[7] (ARI) is essentially the RI that has been adjusted for chance. The RI does not account for those pairs of elements that have the same distribution in both clusterings by chance, whereas the ARI does. The implementation used in this paper has a range of –0.5 to 1, where negative numbers indicate disagreement, scores around 0 pairings purely by chance, and 1 complete agreement between the two clusterings on which pair of elements belongs to the same (or different) cluster.

IV. Results and Discussion

The obtained results are displayed in Table 1. The results have been grouped by question and sorted by cluster purity in a decreasing order. For each of the four selected measures a higher number means a better score. The best results are indicated with bold letters. In all cases, the approach based on the modified XLM-R-12 model yields the best results. The high purity of clusters obtained using the vectors constructed by the finetuned model means that, if a grader were shown answers grouped according to the results of the clustering, they would be shown very similar answers one after the other. It has been shown that such grouping can speed the process of grading up [13], [15]. Furthermore, very few answers need to be used for finetuning, which means the grader would not need to do much grading beforehand.

Interestingly, fastText representations mostly yield better results than the non-finetuned cross-lingual models, even though the models have a greater capability of encoding context. This could be a result of Croatian being a low resource language [21], [22], which makes it less likely the model has seen many examples that fall within the CS domain during its pre-training phase. Furthermore, the fairly consistent last place the XLM-R-24 takes may be due to the size of the model and the aforementioned scarcity of examples.

Table 1: Cluster quality scores for all methods used in this paper.

Answer	Model	Cluster purity	NMI	RI	ARI
Q1	XLM-R-12-modified	**0,647887324**	**0,550325943**	**0,741247485**	**0,432060567**
	FastText	0,563380282	0,489899417	0,687323944	0,319812449
	XLM-R-12	0,535211268	0,488588322	0,669617706	0,31468908
	XLM-R-24	0,507042254	0,470907847	0,655130785	0,30095057
Q2	XLM-R-12-modified	**0,732394366**	**0,624871623**	**0,798390342**	**0,559145192**
	FastText	0,577464789	0,48728793	0,675251509	0,327944656
	XLM-R-12	0,535211268	0,488588322	0,669617706	0,31468908
	XLM-R-24	0,507042254	0,470907847	0,655130785	0,30095057
Q3	XLM-R-12-modified	**0,774647887**	**0,607358355**	**0,773843058**	**0,561388136**
	FastText	0,563380282	0,489899417	0,687323944	0,319812449
	XLM-R-12	0,535211268	0,488588322	0,669617706	0,31468908
	XLM-R-24	0,507042254	0,470907847	0,655130785	0,30095057
Q4	XLM-R-12-modified	**0,746478873**	**0,652056433**	**0,826156942**	**0,60922951**
	FastText	0,577464789	0,48728793	0,675251509	0,327944656
	XLM-R-12	0,535211268	0,488588322	0,669617706	0,31468908
	XLM-R-24	0,507042254	0,470907847	0,655130785	0,30095057
Q5	XLM-R-12-modified	**0,774647887**	**0,731058037**	**0,889336016**	**0,743018779**
	FastText	0,591549296	0,499113707	0,687323944	0,347823003
	XLM-R-12	0,535211268	0,488588322	0,669617706	0,31468908
	XLM-R-24	0,507042254	0,470907847	0,655130785	0,30095057
Q6	XLM-R-12-modified	**0,690140845**	**0,591833147**	**0,756539235**	**0,481332382**
	XLM-R-24	0,61971831	0,494363828	0,686519115	0,382209113
	FastText	0,577464789	0,482160491	0,679275654	0,32754582
	XLM-R-12	0,535211268	0,425642942	0,641448692	0,279889356

[7] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html#sklearn.metrics.adjusted_rand_score

The results indicate that researchers have an important decision to make when constructing (semi-)automatic answer grading solutions. If it is important for the solution to be general and easily transferrable to a data set with different content, fastText and similar static encoding solutions are a better option. However, it is important to note it is unlikely such a solution will provide state-of-the-art results, since it cannot encode the subtleties of the specific content. On the other hand, if the goal is to develop a solution that would give better results, a finetuned cross-lingual model is the way to go. Naturally, this sort of solution is highly specific to the problem and cannot be used on different content (at least not meaningfully). This trade-off currently seems inescapable.

V. Conclusion

In this paper, pre-scoring clustering approaches based on several pre-trained cross-lingual representation models and an additionally modified one have been explored for short answer grading. All models were evaluated on a dataset containing textual answers in Croatian to six questions related to cyber security topics. The obtained results are promising and show improvements regarding the cluster purity, normalized mutual information, Rand index, and adjusted Rand index measures. The improvements are dependent on the properties of each question. The results also indicate there is a trade-off between the power of generalization and high-quality clusters (clusters that are pure), since better results are achieved by finetuning the model for a specific context.

References

[1] S. Basu, C. Jacobs, and L. Vanderwende, 'Powergrading: a Clustering Approach to Amplify Human Effort for Short Answer Grading', Trans. Assoc. Comput. Linguist., vol. 1, pp. 391–402, 2013.

[2] A. Horbach and T. Zesch, 'The Influence of Variance in Learner Answers on Automatic Content Scoring', Front. Educ., vol. 4, no. April, 2019.

[3] J. Lun, J. Zhu, Y. Tang, and M. Yang, 'Multiple data augmentation strategies for improving performance on automatic short answer scoring', AAAI 2020 - 34th AAAI Conf. Artif. Intell., pp. 13446–13453, 2020.

[4] J. D. Karpicke, 'Retrieval-Based Learning: Active Retrieval Promotes Meaningful Learning', Curr. Dir. Psychol. Sci., vol. 21, no. 3, pp. 157–163, 2012.

[5] B. Wisniewski, K. Zierer, and J. Hattie, 'The Power of Feedback Revisited: A Meta-Analysis of Educational Feedback Research', Front. Psychol., vol. 10, no. January, pp. 1–14, 2020.

[6] K. Scalise and B. Gifford, 'Computer-based assessment in E-learning: A framework for constructing "intermediate constraint" questions and tasks for technology platforms', J. Technol. Learn. Assess., vol. 4, no. 6, pp. 3–44, 2006.

[7] R. C. Anderson and W. B. Biddle, 'On asking people questions about what they are reading', Psychol. Learn. Motiv. - Adv. Res. Theory, vol. 9, no. C, pp. 89–132, 1975.

[8] A. Horbach and M. Pinkal, 'Semi-supervised clustering for short answer scoring', Lr. 2018 - 11th Int. Conf. Lang. Resour. Eval., pp. 4065–4071, 2019.

[9] A. Horbach and A. Palmer, 'Investigating active learning for short-answer scoring', Proc. 11th Work. Innov. Use NLP Build. Educ. Appl. BEA 2016 2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL-HLT 2016, pp. 301–311, 2016.

[10] T. M. Tashu, D. Szabó, and T. Horváth, 'Reducing annotation effort in automatic essay evaluation using locality sensitive hashing', Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 11528 LNCS, pp. 186–192, 2019.

[11] A. Horbach, A. Palmer, and M. Wolska, 'Finding a tradeoff between accuracy and Rater's workload in grading clustered short answers', Proc. 9th Int. Conf. Lang. Resour. Eval. Lr. 2014, pp. 588–595, 2014.

[12] M. Mieskes and U. Pado, 'Work Smart - Reducing Effort in Short-Answer Grading', Proc. 7th Work. {NLP} Comput. Assist. Lang. Learn., vol. 2018, no. Nlp4call, pp. 57–68, 2018.

[13] U. Pado and C. Kiefer, 'Short Answer Grading: When Sorting Helps and When it Doesn't', Proc. 4th Work. NLP Comput. Assist. Lang. Learn. NODALIDA, pp. 42–50, 2015.

[14] T. Zesch, M. Heilman, and A. Cahill, 'Reducing annotation efforts in supervised short answer scoring', 10th Work. Innov. Use NLP Build. Educ. Appl. BEA 2015 2015 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL-HLT 2015, pp. 124–132, 2015.

[15] M. Wolska, A. Horbach, and A. Palmer, 'Computer-assisted scoring of short responses: The efficiency of a clustering-based approach in a real-life task', Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8686, pp. 298–310, 2014.

[16] M. Heilman and N. Madnani, 'The impact of training data on automated short answer scoring performance', 10th Work. Innov. Use NLP Build. Educ. Appl. BEA 2015 2015 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL-HLT 2015, pp. 81–85, 2015.

[17] S. J. Greenhill, 'Levenshtein distances fail to identify language relationships accurately', Comput. Linguist., vol. 37, no. 4, pp. 689–698, 2011.

[18] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, 'BERT: Pre-training of deep bidirectional transformers for language understanding', NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., vol. 1, no. Mlm, pp. 4171–4186, 2019.

[19] Y. Liu et al., 'RoBERTa: A Robustly Optimized BERT Pretraining Approach', no. 1, 2019.

[20] A. Conneau et al., 'Unsupervised cross-lingual representation learning at scale', Proc. Annu. Meet. Assoc. Comput. Linguist., pp. 8440–8451, 2020.

[21] F. Pedregosa et al., 'Scikit-learn: Machine learning in Python', J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.