Assessing the Practical Benefit of Automated Short-Answer Graders

Ulrike Padó

Hochschule für Technik Stuttgart, Germany
ulrike.pado@hft-stuttgart.de

Abstract. Short-Answer Grading (SAG) is a task where student answers to open questions are automatically graded with the support of Natural Language Processing and Machine Learning (ML), saving manual effort. Two main challenges remain in small-scale testing scenarios: (1) ML models work best given large amounts of manually graded training instances, and (2) published evaluation results for pre-trained models do not translate well to new data sets, making automated grading intransparent for teachers and students. We present a grader evaluation workflow that teachers can use for their individual situation.

Keywords: Short-Answer Grading · Evaluation · Reliability

1 Introduction

Short answer questions (also called constructed response questions) are a popular task on written exams. Students respond with about one to three sentences in their own words, which makes it easier for teachers to understand their reasoning and spot misconceptions. However, manual grading is time-consuming, especially if frequent feedback through formative testing is desired. Supporting this process with automated grade predictions is the goal of Short-Answer Grading (SAG) [1]. Human involvement can thus be limited to reviewing rather than grading from scratch [9], or by focusing grading effort where it is most needed [6].

Automated methods need training data. However, in many classroom and self-learning settings, there are few existing annotated answers. Recently, transfer learning for Transformer-based models like BERT [3] allows the use of large amounts of un-annotated data to infer a robust language model in pre-training before switching to fine-tuning on a smaller data set for a specific task [5, 2].

However, at the moment, it is unknown how well these results transfer to small-scale testing: The standard literature data sets, at a size of several thousand answers, are small in the context of ML, but still large in the context of small-scale testing. Additionally, prediction quality of ML models deteriorates for new data sets.

We propose a workflow and suggest decision-making parameters for the evaluation of an existing automated grader for a specific classroom. The workflow allows teachers to make an informed decision about how to integrate automated grading support and helps teachers and students understand the performance and limitations of the tool in use. Tan et al. [8] conceptualize the desired qualities of an automated grader as its *reliability* as defined by IEEE: ”the degree to which a system, product or component performs specified functions under specified conditions for a specified period of time” and propose a similar development cycle with many stakeholders and multiple iterations before system deployment. Our use case is much more constrained: We look at the situation-specific evaluation of an off-the-shelf tool for a specific usage scenario by the end-users, which is a linear process with a defined end since fewer modifications to the model are possible.

2 Evaluation Process and Worked Example

In a typical small-scale grading setting, a teacher who wishes to use automated grading will have access to publicly available ML models, and very limited amounts of manually graded data (for example from a recent test) for evaluation. Starting from this position, we propose the following steps:

1. **Define** the requirements for reliability in the current use case
2. **Collect** a set of manually graded test data
3. **Fine-tune** an automated grader for SAG
4. **Analyze** the automated grader's performance
5. **Decide** on how to use the approach

We will now demonstrate these steps using the Huggingface BERT$_{MNLI}$ model as a grader and the SemEval-2013 Beetle test data [4] as manually graded data set.

Defining Requirements

1. **Minimal grading error** We will accept automated labelling error of up to 15%, which has been deemed acceptable in the past for published SAG data [6]. That means we require a grading Accuracy (overall percentage of correctly predicted labels) of at least 85%.

   Another important aspect of grading error is its distribution (showing over-strictness or over-lenience). The model tendency can be measured by looking at the grade labels' Precision separately. (Precision measures how many predictions of a specific label were actually correct, that is, how trustworthy the grader's label predictions are.) In our worked example, we will accept over-lenience, but not over-strictness.
2. **Workload reduction** is the driving factor behind the use of automated graders. As a point of reference, Vittorini et al. [9] report a grading time reduction of about 40% by their approach.

Model	Accuracy	Precision	Recall	F_1
overall	62.5	66.2	62.5	64.3
correct	**75.9**	53.8	75.9	63.0
incorrect	52.8	**75.1**	52.8	62.0

Table 1. Performance of the SEB-tuned model on the Beetle test set (overall/by label).

Data Collection In the context of ML, annotated data is needed in two places: For model training (here, the fine-tuning step of the Transformer-based learner) and for model evaluation. For the fine-tuning step, several thousand manually graded answers are realistically required. Since this is more than will be available to most teachers, publicly available data can be used instead. In our example, this is the SemEval-2013 SciEntsBank training data, containing ca. 5000 answers [4]. The corpus is roughly similar, but different from the source of our test data.

The test set should be as large as possible to make it robust to chance fluctuations; a size in the hundreds of answers is realistic. The Beetle *2-way unseen questions test set* used in our example consists of 9 questions and a total of 819 answers, and differs from the training data in similar ways as field data would differ from a literature data set.

Fine-Tuning adapts the BERT prediction model more closely to the SAG task. Additionally pre-training Transformers for a SAG-related task first tends to further increase performance [2]. Two natural choices of related tasks are Natural Language Inference (using the GLUE MNLI – Multi-Genre Natural Language Inference – data) and Paraphrasing (using the GLUE MRPC – the Microsoft Research Paraphrase Corpus). We therefore compare the $\text{BERT}_{base\_uncased}$ model to BERT_{MNLI} and BERT_{MRPC}[1] on SEB development data (10% of the training data, randomly sampled) after fine-tuning on the remainder of the SEB training data. BERT_{MNLI} outperforms the other models at F_1[2] $= 84.56$ (BERT_{MRPC}: 83.13, BERT_{base}: 83.87).

This model will be used for evaluation below. On the SEB *2-way unseen questions* test set, it performs comparably to the most recent directly comparable literature at F_1 of 73.5 compared to 74.8 [7].

Analysis Table 1 shows our results after evaluating on the 819 manually graded Beetle answers. The model loses 11 points F_1 score in the transfer from the SEB to the Beetle corpus, underscoring that every data set needs individual evaluation. Overall model Accuracy is far below our requirement of 85%.

We now investigate grader tendencies: The automated grader errs strongly towards lenience and misses almost 50% of incorrect answers at Recall_{incorr}=52.8. In consequence, Precision_{corr} is only 53.8. However, the Precision of *incorrect*

[1] All models are available on huggingface.co.

[2] F_1 is the harmonic mean of Precision (reliability of predictions) and Recall (percentage of instances correctly identified).

predictions is 75.1, which means that this label is generally reliable when assigned.

Decision on Usage Given our requirements, a stand-alone use of the automated grader is not acceptable. However, if a human grader accepts all machine-labelled *incorrect* grades, manual grading workload will fall from 819 answers (the whole data set) to 485 (the answers labelled *correct* only) – a reduction of 40.8% and the workload reduction we expect.

This approach would eliminate most of the automated grader's labelling error: If we generously assume that the human grader will always assign the right label, the human-graded 485 answers plus 0.751*334 answers (the answers correctly machine-graded as *incorrect*) are now graded without error. This is 89.8% of all answers, or a grading error rate of 10.2%. This is in the acceptability range we defined above.

However, we know that any remaining errors in this setting are to the detriment of the students. Alternatively, the answers machine-labelled as *incorrect* can additionally be reviewed manually, which is still faster than assigning grades from scratch [9].

In sum, we have shown how analysis in context yields useful information to inform decisions on model usage. The technical effort needed for implementation runs to a few hours of Python coding, thanks to publicly available model libraries and data sets.

References

1. Burrows, S., Gurevych, I., Stein, B.: The eras and trends of automatic short answer grading. IJAIED **25**, 60–117 (2015)
2. Camus, L., Filighera, A.: Investigating transformers for automatic short answer grading. In: AIED Proceedings. pp. 43–48. LNCS (2020)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the NAACL:HLT. pp. 4171–4186 (Jun 2019)
4. Dzikovska, M., Nielsen, R., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., Clark, P., Dagan, I., Dang, H.T.: SemEval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In: Proceedings of SemEval 2013. pp. 263–274 (Jun 2013)
5. Ghavidel, H.A., Zouaq, A., Desmarais, M.C.: Using BERT and XLNET for the automatic short answer grading task. In: Proceedings of CSEDU. pp. 58–67 (2020)
6. Mieskes, M., Padó, U.: Work smart - reducing effort in short-answer grading. In: Proceedings of the 7th workshop on NLP for CALL. pp. 57–68 (Nov 2018)
7. Saha, S., Dhamecha, T.I., Marvaniya, S., Sindhgatta, R., Sengupta, B.: Sentence level or token level features for automatic short answer grading?: Use both. In: AIED Proceedings. pp. 503–517. LNCS (2018)
8. Tan, S., Joty, S., Baxter, K., Taeihagh, A., Bennett, G.A., Kan, M.Y.: Reliability testing for natural language processing systems. In: Proceedings of ACL-IJCNLP. pp. 4153–4169 (2021)
9. Vittorini, P., Menini, S., Tonelli, S.: An AI-based system for formative and summative assessment in Data Science courses. IJAIED **31**, 159–185 (2021)