Similarity-Based Content Scoring - How to Make S-BERT Keep Up With BERT

Marie Bexte and **Andrea Horbach** and **Torsten Zesch**
Research Cluster D^2L^2 - Digitalization, Diversity
and Lifelong Learning. Consequences for Higher Education.
FernUniversität in Hagen, Germany

Abstract

The dominating paradigm for content scoring is to learn an instance-based model, i.e. to use lexical features derived from the learner answers themselves. An alternative approach that receives much less attention is however to learn a similarity-based model. We introduce an architecture that efficiently learns a similarity model and find that results on the standard ASAP dataset are on par with a BERT-based classification approach.

1 Introduction

Most work on automatic content scoring follows an *instance-based* approach, where the input is a single student answer and the output is its score (Horbach and Zesch, 2019). In contrast, *similarity-based* approaches compare a student answer with a - or a set of - reference answers. The two approaches have rarely been compared directly, see Sakaguchi et al. (2015) as the rare exception, who found that instance-based methods outperform similarity-based ones. However, what many situations for which similarity-based methods are proposed have in common is that very little or no training data is available for an individual prompt.

In the following discussion of previous work, we restrict ourselves to those similarity-based approaches. An early example of using reference answers and a similarity function is c-rater (Leacock and Chodorow, 2003). Other examples of pre-neural similarity-based approaches use Wordnet-based and dependency graph alignment measures (Mohler and Mihalcea, 2009; Mohler et al., 2011). Similar approaches have been used for reading comprehension questions (Bailey and Meurers, 2008; Meurers et al., 2011) or scoring history exams (Rodrigues and Oliveira, 2014). The SemEval2013 Student Response Analysis Task (Dzikovska et al., 2013) links content scoring with recognizing textual entailment. Due to the task setup (large number of individual questions with relatively few individual training data per prompt), some participants of the task used similarity-based methods for scoring (Heilman and Madnani, 2013), including methods for recognizing (partial) textual entailment (Levy et al., 2013a,b).

In recent years, neural similarity-based scoring models have been developed. Gomaa and Fahmy (2019) use pretrained skip-thought vectors and learn a logistic classifier over the component-wise product and absolute difference vectors. Schneider et al. (2022) report promising results on a not-publicly-available dataset by learning embeddings for question-answer-pairs and utilize cosine similarity as distance metric.

While the work by Sakaguchi et al. (2015) seems to indicate that similarity-based approaches cannot compete with instance-based ones, such a comparison has so far to our knowledge not been made using powerful neural architectures.

We thus propose a method, where a pretrained Sentence-BERT (S-BERT) model is fine-tuned on answer pairs and then used in a knn-fashion to assign a score to a new learner answer based on the similarity to the already labeled ones.

We present this approach in the next section. Our code is publicly available here: https://github.com/mariebexte/s-bert-similarity-based-content-scoring.

2 Similarity-based Approach

In our similarity-based approach, we learn and apply a similarity function between reference answers and learner answers (see Figure 1). In the simplest case, we use the *all-MiniLM-L6-v2* pre-trained sentence-BERT model (Reimers and Gurevych, 2019) as is to encode answers (**S-BERT-orig**). Alternatively, we finetune the model using answer pairs from our dataset as input (**S-BERT-finetune**). For doing this, we use a CosineSimilarityLoss and a BinaryClassificationEvaluator. We consider answer pairs with the same human score as positive instances (i.e. highly similar) while we consider

Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022), pages 118 - 123

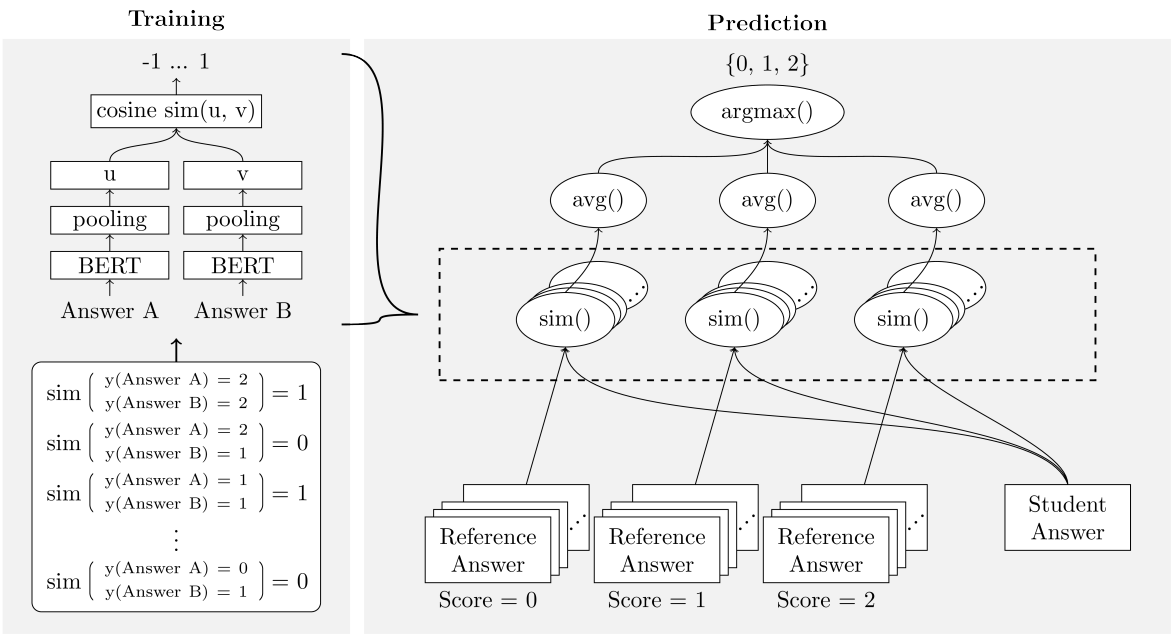

Figure 1: Visual description of our similarity-based approach when using the AVG-strategy to determine predictions.

pairs with different scores as dissimilar. To reflect this (dis-)similarity, we assign positive instances a similarity score of 1 and negative instances a score of 0. We thus refrain from encoding the distance between the number of points with different levels of similarity (ASAP prompts are scored on a range from 0 to 2 or 0 to 3 points), i.e. both pairing an answer that received 0 points with one that received 1 or one that received 2 points gives the same similarity label of 0. This is beneficial when models trained on one prompt are used to score answers to another prompt that has a different number of outcomes.

In the prediction step, we apply the S-BERT model to encode each answer as a single dense vector. We compute the cosine similarity between an answer and each available reference answer. Adopting a knn-inspired approach, we then take either the label of the closest reference answer (MAX) or the label of the group of answers with the same score that has the highest average similarity (AVG). In both cases, the number of necessary comparisons is determined by the number of reference answers. Therefore, scoring more answers will always just require comparing them to this fixed amount of reference answers, whose embeddings can be precomputed.

Note that in our experiments the same data was used to both fine-tune the similarity metric and for comparison in the prediction step. However, if runtime at test time is an issue, one could of course use fewer instances for the comparison than those used for fine-tuning. In our experiments, we observed that using a subset of just 60 of the over 1000 reference answers during inference lead to only a minor drop of QWK .01 in performance.

Do also note that, while we learn similarities between reference answers during training and use the same reference answers when later scoring answers, this does not reflect an inappropriate data leak between training and testing, as we are still scoring previously unseen answers.

3 Experimental Setup

We use the following setup to compare our approach against instance-based state-of-the-art systems. All results are averaged over five runs.

Instance-based Baselines To establish a baseline for instance-based classification, we train one supervised classifier per prompt. We use a Logistic Regression (LR) classifier in standard configuration (class_weight='balanced', max_iter=1000) with token uni- to trigram features provided through Scikit-learn as an instance of an explainable shallow learning classifier. As an instance for a neural classifier, we use a BERT model based on the Huggingface implementation.[1] We train for 6 epochs with batch size 16, CrossEntropyLoss, and Adam

[1] https://huggingface.co/bert-base-uncased

optimizer.

Dataset We use the ASAP-SAS dataset from the Kaggle short answer competition[2] containing 10 prompts with around 2,000 answers per prompt. The average answer length of the prompts ranges from 26.5 to 66.2 tokens per answer. Broad prompt topics fall into three categories: *Sciences* (prompts 1, 2 and 10), *English Language Arts* (*ELA*; prompts 3, 4, 7, 8 and 9) and *Biology* (prompts 5 and 6). We use this topic information later to check whether training on a different prompt from the same topic is beneficial. The dataset contains scoring rubrics but no specific set of reference answers for the individual scores. Whenever we talk about reference answers, we mean answers drawn from the pool of training data.

Data Split and Evaluation Method We randomly chose 10% of the answers for each prompt as testing data and report results as quadratically weighted kappa (QWK). As the amount of human-scored data needed to train a classifier is a crucial factor determining the costs of automatic scoring approaches, we compare two setups. *Limited data* contains 60 learner answers sampled from the full training data set in a way that all scores are equally represented. Mimicking a strategy where clear reference answers are provided to human annotators, we only select answers where both human annotators agreed on the score. *Full data* in contrast consists of the whole training set.

For our similarity-based approach we in the *limited data* setting use 48 of the 60 answers for training and the remaining 12 for validation. Within both of these sets, we build all possible pairs of answers, meaning that we end up with 2,256 training and 132 validation examples. As described in the previous section and visualized in Figure 1, these pairs are assigned a similarity score of 1 or 0, depending on whether they received the same or a different number of points.

For the *full data* setting, we randomly select 100 answers for validation and leave the rest for training. We pair every training (validation) answer with 10 other answers per score to create training pairs. Depending on the number of different possible scores of a prompt, this gives around 3,000 validation and between 40,000 and 60,000 training examples.

[2]https://www.kaggle.com/c/asap-sas

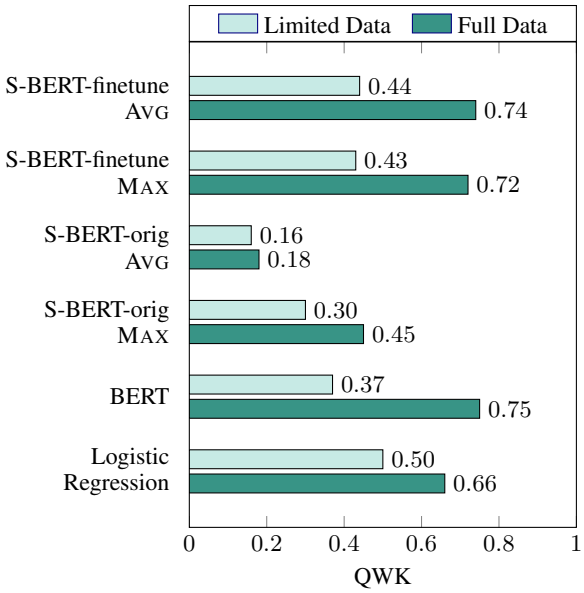

Figure 2: QWK averaged over all prompts (after Fisher-Z transformation), either using just 60 instances (limited data) or 90% of the ASAP data (full data).

4 Results

Figure 2 shows the comparison between the instance-based baseline and the similarity-based approach on both limited and full data. Note that the different amounts of training data also mean that there are different amounts of reference answers.

Comparing S-BERT-orig and S-BERT-finetune reveals that finetuning is highly beneficial. MAX performs much better than AVG with the pretrained model, perhaps due to just one similar enough answer sufficing for the MAX strategy to arrive at the correct classification outcome. For the finetuned models the performance difference between the two is much smaller, with AVG even giving slightly better results than MAX.

In the *limited data* setting, BERT is not able to learn a sufficiently good model from the few training instances. LR performs best in this setting, beating our S-BERT-finetune by QWK .06. In the *full* train setting, S-BERT-finetune is on par with BERT when using AVG to determine predictions, with both models outperforming LR.

Variance between Prompts Table 1 breaks results down to individual prompts. We see that performance is largely prompt-dependent and that there is no one-best model across all prompts. While LR gives the overall best performance when using limited data, there are prompts where S-

Topic	Prompt	Limited Train				Full Train			
		LR	BERT	S-BERT-finetune		LR	BERT	S-BERT-finetune	
				MAX	AVG			MAX	AVG
Science	1	.58	.26	.50	.54	.82	.89	.84	.88
	2	.52	.18	.34	.12	.67	.77	.79	.78
	10	.45	.53	.37	.47	.66	.70	.67	.74
ELA	3	.42	.44	.36	.38	.70	.67	.67	.72
	4	.54	.44	.49	.54	.70	.73	.67	.69
	7	.19	.15	.41	.29	.51	.72	.72	.74
	8	.49	.40	.35	.36	.43	.66	.58	.58
	9	.61	.53	.48	.53	.64	.71	.68	.69
Biology	5	.47	.35	.49	.48	.73	.77	.77	.77
	6	.66	.36	.52	.61	.66	.79	.72	.72

Table 1: QWK per prompt, either using 3-4% (60 instances, limited train) or 90% of the ASAP data (full train).

BERT-finetune performs better than LR, indicating that there are some prompts for which using a similarity-based approach is more suitable than for others. For prompts 2, 5 and 7, BERT gives rather low QWK on limited data, which is outperformed by S-BERT-finetune. While BERT gives much better performance on these prompts in the full data setting, it is again outperformed by or on par with S-BERT-finetune.

Cross-prompt Evaluation One of the assumed benefits of similarity-based scoring approaches is that they generalize better between prompts and are thus often used for prompt-independent scoring (Meurers et al., 2011; Mohler et al., 2011; Mohler and Mihalcea, 2009; Dzikovska et al., 2013). We hypothesize that using a model from the same topic (Science, Biology, ELA) will work better than using a model from a different topic. Table 2 reports results for models trained on a different prompt than the test data. In doing this, we use the larger number of training pairs from the full data setting to train a model and evaluate it with the smaller number of reference answers from the limited data setting.

We average across all prompts from the same topic, i.e. the cell *train/science - test/science* contains averaged results, where a model has been trained on one science prompt and tested on another science prompt. Results show that only for Biology prompts training on the same prompt is clearly beneficial as compared to training on other prompts. However, it still is much worse than fine-tuning directly on a single prompt. For example, the average QWK on the two Biology prompts is over 0.70 for the fine-tuned results, while it is only half of that in the cross-prompt setting. For the other topic areas (Science, ELA) the cross-prompt results are even worse.

Train	Test					
	Science		Bio		ELA	
	MAX	AVG	MAX	AVG	MAX	AVG
Science	.30	.29	.18	.09	.23	.16
Bio	.31	.18	.35	.40	.27	.17
ELA	.28	.16	.26	.19	.22	.16
S-BERT-orig	.34	.23	.24	.13	.30	.13

Table 2: Average QWK (after Fisher-Z transformation) for training S-BERT on a prompt from one topic group and testing on another prompt from the same/a different group.

Another cross-prompt setting would be to use the pretrained S-BERT-orig model as a zero-shot classifier (cf. the last line in Table 2). Results are in a similar ballpark as for the within-topic setting, which means that fine-tuning on one prompt and transferring to a similar one does not work better than not fine-tuning at all. Thus, it is necessary to learn a prompt-specific similarity function to arrive at reasonable performance levels. Contrary to our hypothesis, a similarity function learned on a different prompt from the same dataset and topic did not work better than using one that was trained on an entirely different dataset and topic.

5 Conclusion

In contrast to earlier work where instance-based methods outperformed similarity-based ones, the study in this paper finds that both paradigms are on par when a neural similarity model has been sufficiently fine-tuned. This seems to indicate that as soon as a similarity metric is complex enough, it incorporates the same capabilities as normally a classifier would. For the practitioner it might make

little difference whether to use labeled instances to train an instance-based classifier or to fine-tune a similarity metric if both are applied in a prompt-specific way. Therefore, the next step in our line of research has to go into the direction of fully comparing the two paradigms, especially with respect to varying the amount of training data as well as exploring other datasets to allow for a better estimation which paradigm is preferable under which conditions.

One step that we already took in this direction was to use the architecture described here for our participation in the NAEP-AS challenge[3], where our generic scoring model won a grand prize. In contrast to the successful application there, our cross-prompt experiments reported here showed results varying tremendously between prompts, hinting that sensible training data selection plays a crucial role. We will explore this further in future work. To foster more work in this area, we make our experimental code publicly available.

6 Acknowledgments

This work was conducted in the framework of the Research Cluster D^2L^2 "Digitalization, Diversity and Lifelong Learning – Consequences for Higher Education" of the FernUniversität in Hagen, Germany (https://e.feu.de/english-d2l2). The work was partially conducted within the KI-Starter project "Explaining AI Predictions of Semantic Relationships" funded by the Ministry of Culture and Science Nordrhein-Westfalen, Germany.

References

Stacey Bailey and Detmar Meurers. 2008. Diagnosing meaning errors in short answers to reading comprehension questions. In *Proceedings of the third workshop on innovative use of NLP for building educational applications*, pages 107–115.

Myroslava O Dzikovska, Rodney D Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa T Dang. 2013. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. Technical report, NORTH TEXAS STATE UNIV DENTON.

Wael Hassan Gomaa and Aly Aly Fahmy. 2019. Ans2vec: A scoring system for short answers. In *International Conference on Advanced Machine Learning Technologies and Applications*, pages 586–595. Springer.

[3]https://github.com/NAEP-AS-Challenge/info

Michael Heilman and Nitin Madnani. 2013. ETS: Domain adaptation and stacking for short answer scoring. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 275–279.

Andrea Horbach and Torsten Zesch. 2019. The influence of variance in learner answers on automatic content scoring. In *Frontiers in Education*, volume 4, page 28. Frontiers.

Claudia Leacock and Martin Chodorow. 2003. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4):389–405.

Omer Levy, Torsten Zesch, Ido Dagan, and Iryna Gurevych. 2013a. Recognizing partial textual entailment. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 451–455.

Omer Levy, Torsten Zesch, Ido Dagan, and Iryna Gurevych. 2013b. UKP-BIU: Similarity and entailment metrics for student response analysis. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 285–289.

Detmar Meurers, Ramon Ziai, Niels Ott, and Janina Kopp. 2011. Evaluating answers to reading comprehension questions in context: Results for German and the role of information structure. In *Proceedings of the TextInfer 2011 Workshop on Textual Entailment*, pages 1–9.

Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 752–762.

Michael Mohler and Rada Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 567–575.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Fátima Rodrigues and Paulo Oliveira. 2014. A system for formative assessment and monitoring of students' progress. *Computers & Education*, 76:30–41.

Keisuke Sakaguchi, Michael Heilman, and Nitin Madnani. 2015. Effective feature integration for automated short answer scoring. In *Proceedings of the 2015 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pages 1049–1054.

Johannes Schneider, Robin Richner, and Micha Riser. 2022. Towards trustworthy autograding of short, multi-lingual, multi-type answers. *arXiv preprint arXiv:2201.03425*.