Explaining transformer-based models for automatic short answer grading

Andrew Poulton
andrew.poulton@alefeducation.com
Alef Education
UAE

Sebas Eliëns
sebas.eliens@gmail.com
Dase, Alef Education
UAE

ABSTRACT

Over recent years, advances in natural language processing have brought ever more advanced and expressive language models to the world. With open-source implementations and model registries, these state-of-the-art models are freely available to anyone, and the successful application of transfer learning has meant benchmarks on previously difficult tasks can be beaten with relative ease.

In this regard, Automatic Short Answer Grading (ASAG) is no different. Unfortunately, an infallible ASAG system is beyond the reach of current models, and so there is an onus on any ASAG implementation to keep a human in the loop to ensure answers are being accurately graded. To assist the humans in the loop, one may apply various *explainability methods* to a model prediction to give clues as to why the model came to its conclusion. However, amongst the many available models and explainability techniques, which ones provide the best accuracy and most intuitive explanations?

This work proposes a framework by which this decision can be made, and assesses several popular transformer-based models with various explainability methods on the widely used benchmark dataset from Semeval-2013.

CCS CONCEPTS

• **Computing methodologies → Information extraction**.

ACM Reference Format:
Andrew Poulton and Sebas Eliëns. 2021. Explaining transformer-based models for automatic short answer grading. In *2021 5th International Conference on Digital Technology in Education (ICDTE 2021), September 15–17, 2021, Busan, Republic of Korea.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3488466.3488479

1 INTRODUCTION

New technologies and developments have resulted in more and more learning taking place using digital technologies, and the ongoing Covid-19 pandemic has given another dramatic impulse to this digitization trend in education. Evaluating the student and providing feedback has traditionally been the job of a teacher or a tutor, but with the rise of digital technologies demand for automated systems with such capabilities is rising as well.

Automatic Short-Answer Grading (ASAG) [1] is the task of automatically grading student answers to questions which are open, but require a short, relatively well-defined answer. As such it strikes a balance between more open test formats which asks more creativity and probe higher order skills of the student, and closed forms such as multiple choice or filling in a missing word which are easy to grade automatically.

Contexts in which ASAG capabilities are likely to be desirable include online learning platforms, large scale tests, and automated dialogue based tutoring systems, which have shown promising results in helping students achieve mastery [2–5]. Recent developments in deep learning and natural language processing (NLP) have made building a custom ASAG system with a performance comparable with state-of-the-art relatively cheap and easy. Crucial in this respect has been the development of large language models such as BERT [6], and the widening application of transfer learning [7].

While deep-learning based approaches are highly successful, approaching or sometimes even surpassing human performance, they suffer from a lack of interpretability of *how* these results are reached. Depending on the context, a lack of explainability can be a serious practical or ethical drawback.

In the context of ASAG there are at least three problems that may be caused by a lack of interpretability: First of all, as a matter of ethics it is likely that the system should provide some feedback alongside the mere grading, for example because the student's future depends on the test result or to help the student understand what they can improve. More descriptive feedback is viewed as more useful by students than a mere grade [8]. Secondly, since the prediction of any system will be imperfect (including human grading) there should always be a mechanism to gain trust in the system and evaluate and correct misclassifications. Every teacher knows that after any test they can expect students knocking on their door to discuss their grade. But how would an automated system argue about its decision? Thirdly, there may be ways to "fool" the ASAG system by adversarial inputs [9], i.e. one has to be careful about cheating. All these issues require for a human to be in the loop.

In this work we explore the recent developments of explainability techniques that can help the human in the loop, and could possibly fully automate grading in the future. The scenario we envision is the following: Suppose one is interested in designing an ASAG system, for instance as part of an EdTech product, and has access to an appropriate dataset and a GPU for training. The ready availability of pretrained models such as BERT should make it easy to reach results that are close to state-of-the-art by fine tuning it on the specific task

[1] Also known as Student Response Analysis (SRA) [1].

ICDTE 2021, September 15–17, 2021, Busan, Republic of Korea
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8499-5/21/09.
https://doi.org/10.1145/3488466.3488479

and dataset (by following [10] or [11], for instance). Explainability methods [2] can also be added using open-source implementations and can help with addressing the challenges of feedback, trust and adversarial inputs. There are however choices to be made in selecting the specific model and explanability methods that gives the best performance (and indeed what constitutes good performance in the ASAG context). Presenting the experimentation needed to make these choices is the main aim of this paper.

2 RELATED WORK

There is large body of literature, related to ASAG, the use of NLP in education in general, langauge models and explainability methods. To keep things concise we review only works that address transfer learning, pretrained language models and contextualized word embeddings for ASAG specifically, as well as works that do a comparative assessments of explainability methods in comparible tasks such as text classification.

Several recent studies have investigated the effectiveness of transfer learning using pre-trained language models in ASAG tasks. In [10], results are reported that are obtained by fine tuning a BERT model (BERT-base-uncased) on the SemEval-2013 ScientsBank-3way dataset and two psychology domain datasets. The work reports a macro F1 score of 0.720 for the Unseen Answer subtask of the SemEval-2013 while reporting F1 = 0.857 and F1 = 0.822 on the two psychology domain datasets respectively, beating earlier results in the literature.

In [13] the Mohler dataset [14] investigated the use of sentence embeddings from four pretrained language model, an LSTM model (ELMo) and three transformer models (GPT, GPT-2 and BERT), in automatically grading the students answer based on similarity with the reference answer. The study finds that the ELMo embeddings give the best results in their setup.

A notable investigation of explainability methods, similar in spirit to our work, is reported in [15]. An important difference between this research and ours is that the authors focus on text classification, not answer grading. More specifically, they study three down-stream classification tasks, e-SNLI dataset predicting entailment, contradiction or neither between sentence pairs, and two sentiment analysis tasks of IMDB film reviews and tweets respectively.

A study of explainability methods focussed on ASAG has to the best of our knowledge not appeared in the literature.

3 BERT AND FRIENDS

In this section we provide some of the technical background related to transformer-based models that are used in our experimentation. We will be brief. Additional details may be found in the provided references.

3.1 Transformer architecture

The transformer architecture consists of multiple layers each consisting of two sublayers, a multi-headed self-attention (MHSA) layer and a feed-forward (FF) layer.

Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be the input to the l-th layer T^l and suppose there are N heads per layer. Then each head h in T^l is parameterized by four matrices $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V, \mathbf{W}_h^O \in \mathbb{R}^{d\times d_h}$, where $d_h = d/N$. Setting $\mathbf{K} = \mathbf{X}\mathbf{W}_h^K$, $\mathbf{Q} = \mathbf{X}\mathbf{W}_h^Q$, and $\mathbf{V} = \mathbf{X}\mathbf{W}_h^V$, h computes

$$h(\mathbf{X}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}\mathbf{W}_h^{OT}.$$

The outputs from the N heads are computed in parallel, and the final output from the MHSA sublayer is

$$\text{MHSA}^l(\mathbf{X}) = \sum_{h\in T^l} h(X).$$

The feed-forward sublayer is a standard two-layer dense neural network with a gelu activation [16]. Explicitly, we have

$$\text{FF}^l(\mathbf{X}) = (\text{GELU}(\mathbf{X}\mathbf{W}_1 + \mathbf{b})_1)\mathbf{W}_2 + \mathbf{b}_2,$$

where $\mathbf{W}_i, \mathbf{b}_i$ are the weight and bias parameters, respectively.

A residual connection is added to the output of each sublayer, and the result is subjected to layer normalization [17]. For full details, we refer to the original transformer paper [18]. Note that we have described the *encoder* version of the transformer layer, as all models considered are built out of these layers.

3.2 Models

The models we tested represent some of the most commonly used transformer-based models of varying sizes and pre-training paradigms. Specifically, we consider BERT [6], RoBERTa [19], their distilled cousins DistilBert [20] and DistilRoBERTa, and ALBERT [21]. In our experiments all of these models consist of an embedding layer (with optional token type embeddings, see 3.3), multiple transformer encoder layers, and a final linear classifier. We describe the main differences between these models below, but refer the reader to the original papers for detailed information.

BERT was the first transformer-based model to apply the now-familiar process of pre-training a large language model on vast amounts of data, and using the resultant weights as the backbone of a new model targeting more specific tasks (such as short answer grading). BERT was (pre-)trained using a masked language model objective (MLM), together with an auxilliary next sentence prediction (NSP).

RoBERTa tunes this procedure by removing the NSP task altogether (but ensuring only complete sentences are seen during pretraining), dynamically masking the input at training time (rather than statically as a preprocessing step as BERT did), and using larger batch sizes and token vocabulary.

Both DistilBert and DistilRoberta are obtained from their larger cousins via knowledge distillation [22], resulting in much smaller models (with faster inference runtime) with only a fractional loss in performance.

[2] We only consider *intrinsic* explanations - often called *rationales* in the literature - here. These methods aggregate feature-level attributions to the token level, which we interpret as giving each input token a relative importance in the model's predictive reasoning. Producing extrinsic explanations (generating text as explanation) is beyond the scope of this work, but has recently been investigated for other tasks in [12]

ALBERT differs from the previous models by sharing parameters across multiple layers, with a large reduction in model size. ALBERT also replace NSP with *sentence order prediction* (SOP), whose objective is to predict whether two consecutive sentences are shown in forward or reverse order.

Finally, we also consider versions of the above models that have been finetuned on the Stanford Question Answering Dataset (SQuAD 2.0 [23]).

3.3 Input Representation

Each input instance contains a question, a reference answer, and submitted answer (whose correctness is to be determined). These are concatenated together (separated by special tokens), tokenized, and then passed in to the model. We optionally provide information to the model as to which part of the input (question, reference, submitted) a particular token belongs by way of *token type embeddings*[3], which are learned independently to the token embeddings and are summed with them in the embedding layer (before dropout and layer norm is applied). ASAG differs from most tasks previously considered in studies on explainability methods in that much of the input (the question and reference answer) should *not* contribute to the model's prediction, but rather should be used to inform the model why the student answer is correct or not. Correspondingly, this should be reflected in any explanation - the tokens in the student answer are important, not those in the question or reference answer. Token types can be used to provide information to the which tokens should be considered for explanations.

All models tokenize their input using byte pair encoding [24] with the vocabulary being determined during pretraining.

4 EXPLAINABILITY METHODS

For this work, we wish to investigate the efficacy of several popular explainability methods. In this section we provide a brief technical background on each of them. See the references for more details.

In the following, we let $F_c(\mathbf{X})$ denote the probability that some input $\mathbf{X}$ belongs to output class c (in our case, whether or not a student answer is correct) according to the model F.

4.1 Saliency

Saliency [25] computes the gradient of $F_c(\mathbf{X})$ with respect to the input $\mathbf{X}$. By interpreting this as the linear approximation of $F_c(\mathbf{X})$ in a neighbourhood of $\mathbf{X}$ via its Taylor series, this is analogous to interpreting the coefficients in a linear regression model as a measure of feature importance.

4.2 InputXgradient

InputXGradient ("input times gradient", [26]) simply scales the saliency of a feature by its input value.

4.3 Integrated Gradients

Integrated gradients (IG) [27] is an axiomatic approach to feature attribution designed to satisfy several desirable properties. It computes the path integral of F_c from some baseline input $\mathbf{X}'$ to $\mathbf{X}$ along the straight line $(1-\alpha)\mathbf{X}' + \alpha\mathbf{X}$ for $\alpha \in [0, 1]$. IG guarantees that the attributions it produces exactly account for the difference in output (at class c) between the baseline input and the input of interest.

To produce baselines, we simply replace the tokens of the student answer with pad tokens. This effectively simulates comparing a student answer to a blank response.

4.4 Occlusion

Whereas the previous explainability methods are all gradient-based, occlusion [28] represents another common attribution paradigm, *input pertubation*. With occlusion we compute the difference $\left|F_c(\mathbf{X}) - F_c(\hat{\mathbf{X}}_\mathbf{i})\right|$, where $\hat{\mathbf{X}}_\mathbf{i}$ denotes a perturbed version of X with the embedding corresponding to token i set to the origin.

4.5 Gradient SHAP

SHAP (SHapley Additive exPlanation) Values [29] were introduced to unify various existing explanation methods such as LIME [30], DeepLift [31], and layerwise relevance propagation [32] under the umbrella of *additive feature attribution methods* (where the explanation model is linear in the features).

GradientSHAP [29] begins by adding gaussian noise to the input X, sampling a point Y on the straightline path from the perturbed X to a point drawn from some baseline distribution (in our case we use the same baseline as we did for integrated gradients, so this is a singleton distribution). After computing the gradient of $F_c Y$ for a number of samples Y, SHAP values can be approximated by taking the expectation of the gradient multiplied elementwise by Y.

5 METRICS FOR EVALUATING ATTRIBUTIONS

For our experiments, we wanted to simulate the process we expect one would take to productionize an explainable transformer model for ASAG. We hypothesize that a good explanation method should satisfy two key properties: the explanations should agree with expert intuition, and should be robust to similarly performant models trained with different random seeds but from the same starting parameters.

Our framework for evaluating explanation methods is based on that of [15]. Specifically, we use human agreement (HA) to measure the first property, and an adaptation of rationale consistency (RC) to measure the second.

To measure HA on a particular instance, we first annotate words in the student answer that are representative of the target class (correct/incorrect). This gives each token a binary "golden" label identifying whether they help explain why the answer was right or wrong, and we interpret attribution values (scaled to lie $[0, 1]$) as prediction probabilities that a given token is a "gold" token or not. HA is then calculated as the mean average precision (MAP) of the gold labels and the attribution values. HA takes values in $[0, 1]$, and higher is better.

Computing RC is a bit more subtle. We expect two training runs of the same model to provide similar reasoning for producing similar explanations, as this gives us confidence that the explanations provided are a result of the way the model reasons and not

[3] We can add token types for all models except for DistilBert, where adding token types would change the model topology (they are not part of the original implementation).

statistical flukes. Given an instance I, and two attribution maps a_I, b_I produced by an explanation method applied to two models A and B (trained with different seeds on the same dataset with the same starting parameters), the similarity $S(a_i, b_i)$ is simply the cosine similarity $\frac{a_i \cdot b_i}{\|a_i\|\|b_i\|}$. We follow [15] and approximate the reasoning path of a model as the activations after each layer, and so compare the two reasoning paths in A and B by taking the mean absolute difference of the activations across all layers, which we denote $D(A, B, i)$. Finally, we compute RC as the weighted average

$$RC = \frac{\sum_i S(a_i, b_i) \exp(-D(A, B, i)/\mu_D)}{\sum_i \exp(-D(A, B, i)/\mu_D)} \tag{1}$$

where the sum is over instances in the validation set, and μ_D is the average activation difference over the validation set. Note that since for any instance and any model, the attribution map takes non-negative values so $S(a_i, b_i) \in [0, 1]$. This implies that RC takes values in $[0, 1]$ as well, and higher is better.

6 EXPERIMENTS

For our experimentation we used the SemEval-2013 dataset [1], from both SciEntsBank and Beetle sources, restricted to the 2-way task in which student answers are classified as either *correct* or *incorrect*. For evaluation we have used the *unseen answers* part of the test dataset and weighted F1 scores for evaluation. The class imbalance correct/incorrect is 233/307 for SciEntsBank, and 176/292 for Beetle.

6.1 Training and attribution

We trained a total of 13 different models using the Pytorch implementation and pre-trained weights made available through the Huggingface Transformers library [33]. The identifying path and further information of the models can be found through the HuggingFace site [4]. We exclusively used uncased models, e.g. bert-based-uncased, and the latest versions, e.g. albert-base-v2 and albert-large-v2, if more than one version is available[5]. Distilled models and models pretrained on SQuAD 2.0 correspond to base, as opposed to large, versions of the model.

Training was done on a single V100 GPU in batches of size 8 for a maximum 25 epochs. If the best F1 score was not surpassed for five epochs we stopped training early. The model with the best F1 score was kept for analysis of the explainability methods. We used AdamW as optimizer and a linearly-decaying learning rate scheduler with warmup. The maximum learning rate and weight decay were both set to 10^{-5} and we used 1024 warmup steps and 17000 as maximum number of steps.

To get more robust statistics on the performance, each of the models has been trained in three different runs for each of the four cases, i.e. on the SciEntsBank and Beetle data and with and without token type IDs.

Computations of attributions uses the implementations from the Captum package [6].

7 RESULTS AND DISCUSSION

In interpreting the results, we imagine a use case where the development of an ASAG system with explainability component requires one to make a balanced choice taking into account required resources and performance. The performance of trained models is summarized in Table 1, while the evaluation of the explainability methods is presented in Tables 2 and 3.

InputXGrad returns the best results for HA whereas Integrated Gradients performed best for RC (both were the best performing method for both datasets). Saliency places a close second in HA, and a slightly more distant second too for RC. As it is likely that any real-world application for ASAG would favour higher HA over higher RC, we would recommend a choice of either InputXGrad or Saliency, with the latter preferred if RC is important (say, if models are frequently retrained). The fact that both these methods are much more computationally efficient compared to Integrated Gradient is a bonus.

Occlusion performs notably more poorly than the gradient-based methods, especially in RC where it underperforms even the random baseline. GradientShap, interpolating between pertubation-based and gradient-based explanation methods, improves on Occlusion but is beaten by all purely gradient-based methods on all tasks.

In terms of model performance, larger models unsurprisingly have the best F1 scores with ALBERT-large and RoBERTa-large performing best. However, it doesn't appear that better performance on the task results in better explanations, with the distilled versions of BERT and RoBERTa generally outperforming their larger cousins in terms of explanations. We also observe that finetuning on SQuAD 2.0 appears to have a benefit when it comes to the performance of the explanation methods.

In all cases, there is no apparent benefit to training with token types, and given their additional complexity (especially when training, leading to slower convergence) we do not recommend including them.

8 CONCLUSION

We surveyed explainability methods for Automatic Short-Answer Grading by putting them to the test for a large number of transformer-based models. Using two metrics, one to test agreement with human annotations of important words and one testing the stability of the method across different trained versions of the model, we evaluated each combination of model and explainability method and found the following indications: While large models, in particular ALBERT and RoBERTa large, show the best performance on the task as expected, evaluation of the explainability methods seems to favor the smaller models, in particular DistilBERT, as providing intuitive and robust attributions. Moreover, finetuning on the SQuAD 2.0 dataset before further task specific finetuning for ASAG shows some beneficial effect for explainability. Adding token type information to the input did not seem to have a clear effect on either performance of explainability.

[4] https://huggingface.co/models

[5] The HuggingFase model paths for the models we used are: bert-base-uncased, bert-large-uncased, roberta-base, roberta-large, albert-base-v2, albert-large-v2, distilbert-base-uncased, distilroberta-base, twmkn9/distilbert-base-uncased-squad2, deepset/roberta-base-squad2, twmkn9/distilroberta-base-squad2, twmkn9/bert-base-uncased-squad2, twmkn9/albert-base-v2-squad2.

[6] https://captum.ai/

Table 1: Training results. We report the weighted-F1 mean μ and standard deviation σ as $\mu(\pm\sigma)$ over three runs. Note that albert-large did not converge in one of the runs for SciEntsBank with token types and we did not take it into account. Further, we did not train either DistilBert model with token types, so those entries are omitted. The bold and underlined entries indicate the columnwise best and second-best results respectively. Entries have been scaled by a factor of 100 for readability

	SciEntsBank		Beetle	
token type IDs	no	yes	no	yes
bert-base	78.9 (±0.49)	80.7 (±0.34)	88.6 (±0.25)	89.9 (±0.09)
bert-large	79.7 (±0.78)	81.6 (±1.69)	89.7 (±0.95)	89.5 (±0.80)
roberta-base	80.1 (±0.46)	80.7 (±0.16)	**91.2** (±0.64)	90.1 (±0.81)
roberta-large	83.1 (±0.30)	82.6 (±1.19)	90.7 (±0.00)	90.2 (±0.10)
albert-base	81.2 (±1.56)	81.1 (±1.61)	89.9 (±0.61)	89.1 (±0.10)
albert-large	**83.4** (±0.10)	**83.7** (±0.05)	91.2 (±0.16)	**90.5** (±1.79)
distilbert	74.9 (±0.16)		89.1 (±0.46)	
distilroberta	78.9 (±1.16)	80.3 (±1.53)	89.0 (±0.73)	89.3 (±0.50)
distilbert-squad2	80.1 (±0.52)		88.4 (±0.13)	
roberta-squad2	81.4 (±1.52)	81.3 (±0.46)	91.0 (±0.10)	89.7 (±0.53)
distilroberta-squad2	77.7 (±0.90)	78.9 (±0.92)	88.4 (±0.13)	89.7 (±0.28)
bert-squad2	79.3 (±1.21)	79.2 (±0.27)	89.9 (±1.14)	90.1 (±2.02)
albert-squad2	76.8 (±0.14)	79.5 (±1.58)	90.5 (±0.21)	89.2 (±1.02)

Table 2: Evaluation of L2 aggregated attributions methods for SciEntsBank data (TT indicates the use of token type IDs). The bold and underlined entries indicate the columnwise best and second-best results respectively. Entries have been scaled by a factor of 100 for readability

	GradientShap		InputXGrad		IntegratedGrads		Occlusion		Saliency		Random	
	HA	RC	HA	RC	HA	RC	HA	RC	HA	RC	HA	RC
bert-base	56.2	87.3	72.3	91.8	55.1	94.4	60.3	73.3	68.8	92.0	43.2	73.8
bert-base-TT	60.0	85.7	71.6	79.3	65.9	90.8	60.6	65.7	69.3	81.6	43.7	74.2
bert-large	56.6	79.8	71.5	84.5	57.8	86.0	**62.2**	62.6	70.6	82.5	43.7	74.0
bert-large-TT	53.5	88.3	73.2	79.6	55.9	91.0	59.1	60.1	72.5	80.0	41.8	74.3
roberta-base	55.3	86.0	72.9	89.3	61.5	91.8	61.8	79.0	68.8	88.8	44.5	73.8
roberta-base-TT	59.1	80.5	70.5	82.9	61.4	83.2	58.0	79.4	68.1	83.6	43.5	73.7
roberta-large	51.9	54.5	69.5	80.6	52.4	61.3	55.7	69.9	67.7	78.4	45.1	73.7
roberta-large-TT	51.0	53.3	68.4	78.0	52.0	56.1	59.9	67.3	66.6	75.9	45.5	74.3
albert-base	58.6	78.6	66.5	84.8	61.4	81.9	60.4	62.4	67.6	85.9	43.3	73.9
albert-base-TT	67.6	83.7	71.0	78.9	69.9	84.7	56.4	59.2	73.9	79.6	43.2	74.0
albert-large	65.7	78.1	71.9	76.2	69.7	80.3	59.9	58.6	73.5	77.5	**45.8**	74.0
albert-large-TT	62.1	78.7	70.5	78.8	65.0	81.6	58.4	44.7	70.7	79.9	41.6	73.0
distilbert	45.0	**99.3**	66.8	91.7	45.1	**99.8**	44.5	85.9	56.4	93.0	45.3	74.4
distilroberta	66.5	91.6	73.4	86.2	68.3	94.2	62.1	72.9	70.3	86.3	45.5	74.2
distilroberta-TT	68.0	90.9	72.4	86.3	68.7	93.2	60.1	75.5	69.8	86.2	45.3	74.2
distilbert-squad2	49.4	94.3	65.6	**96.4**	49.4	95.1	48.9	**89.4**	62.8	**98.7**	43.5	**74.8**
roberta-squad2	61.6	86.0	74.7	87.9	68.2	89.8	61.7	80.1	**74.1**	87.5	44.6	74.5
roberta-squad2-TT	67.3	86.0	**75.0**	88.3	**73.7**	91.8	60.7	76.7	73.1	87.9	42.8	74.0
distilroberta-squad2	60.3	92.7	70.3	89.2	61.1	95.4	59.7	80.2	67.3	89.4	45.5	74.0
distilroberta-squad2-TT	61.5	92.4	71.2	88.8	64.9	95.5	61.1	80.8	66.1	89.2	39.9	74.4
bert-squad2	55.2	84.6	71.1	89.0	61.5	92.0	58.9	64.6	69.5	89.1	44.9	74.1
bert-squad2-TT	55.9	88.3	72.8	85.8	58.8	91.7	61.3	69.7	70.5	86.2	44.2	74.1
albert-squad2	62.6	86.3	65.7	84.3	65.2	88.5	60.6	50.6	67.5	85.8	42.6	74.4
albert-squad2-TT	**69.9**	84.4	69.9	73.3	69.9	87.3	58.9	47.3	70.7	74.5	43.7	73.9
mean	59.2	83.8	70.8	84.7	61.8	87.4	58.8	69.0	69.0	85.0	43.9	74.1

Table 3: Evaluation of L2 aggregated attributions methods for Beetle data (TT indicates the use of token type IDs). The bold and underlined entries indicate the columnwise best and second-best results respectively. Entries have been scaled by a factor of 100 for readability

	GradientShap		InputXGrad		IntegratedGrads		Occlusion		Saliency		Random	
	HA	RC	HA	RC	HA	RC	HA	RC	HA	RC	HA	RC
bert-base	80.4	88.0	86.9	91.6	72.8	93.7	67.9	70.9	**88.0**	91.8	58.5	72.5
bert-base-TT	58.4	85.6	85.9	84.5	59.4	90.8	72.7	67.5	86.2	84.8	65.1	72.6
bert-large	72.2	82.4	83.5	86.2	81.4	90.0	78.6	73.5	79.8	84.5	63.8	73.1
bert-large-TT	61.2	88.4	81.9	82.8	64.2	90.8	75.2	67.6	76.3	83.1	69.0	73.0
roberta-base	78.1	84.8	85.3	87.9	77.0	89.6	74.1	83.5	81.3	87.3	58.7	73.5
roberta-base-TT	**82.6**	85.4	**88.6**	86.4	**89.2**	90.6	82.5	79.8	85.1	85.9	61.0	73.3
roberta-large	69.5	58.3	79.1	82.7	62.0	64.2	76.8	73.2	78.2	80.3	69.8	73.3
roberta-large-TT	58.9	53.2	83.1	79.9	78.2	62.0	73.7	71.9	81.5	77.2	65.8	73.2
albert-base	78.3	84.2	70.0	80.2	83.4	86.7	**82.9**	44.3	72.7	82.5	63.6	73.4
albert-base-TT	74.8	87.5	71.5	82.5	73.9	91.0	73.5	67.7	81.1	83.7	64.7	73.3
albert-large	73.2	82.3	84.1	80.1	73.1	86.2	79.3	44.7	82.8	82.0	54.9	**75.3**
albert-large-TT	73.5	74.7	81.8	77.4	74.6	84.5	72.1	66.8	83.1	79.5	63.9	73.1
distilbert	67.7	**98.8**	83.2	90.8	71.3	**99.5**	63.2	76.3	80.0	94.5	52.5	73.3
distilroberta	79.2	92.0	86.9	85.4	83.6	94.2	81.8	74.4	84.4	86.0	64.3	73.2
distilroberta-TT	79.3	92.7	87.9	90.0	80.5	95.1	73.5	78.4	87.8	91.3	68.1	73.3
distilbert-squad2	62.4	96.5	83.1	**98.4**	68.9	99.4	63.4	**91.5**	81.5	**99.1**	62.0	72.9
roberta-squad2	76.0	84.6	87.3	89.5	73.0	88.1	68.6	80.4	83.0	88.7	64.9	73.1
roberta-squad2-TT	78.2	85.2	85.8	88.0	83.0	90.9	75.1	82.7	82.9	87.1	67.2	73.4
distilroberta-squad2	71.9	93.5	86.1	90.5	73.7	95.4	76.1	79.9	79.9	91.2	**70.0**	73.1
distilroberta-squad2-TT	74.2	91.1	88.4	77.9	75.9	92.2	76.2	67.3	86.4	79.4	61.4	72.8
bert-squad2	75.8	89.4	84.4	91.6	84.3	95.2	73.2	76.8	83.7	91.0	53.1	73.1
bert-squad2-TT	65.8	86.6	83.3	81.5	71.7	90.6	79.2	67.3	83.4	80.5	54.9	73.2
albert-squad2	80.4	90.1	84.2	83.5	80.4	91.8	77.2	66.3	86.5	85.4	67.0	73.3
albert-squad2-TT	79.1	85.9	82.5	77.5	86.2	89.8	74.3	54.8	84.8	79.3	60.1	72.5
mean	73.0	85.1	83.5	85.3	75.9	89.2	74.6	71.2	82.5	85.7	62.7	73.2

REFERENCES

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

[2] Sidney D'mello and Art Graesser. Autotutor and affective autotutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(4):1–39, 2013.

[3] Patricia Albacete, Pamela Jordan, and Sandra Katz. Is a dialogue-based tutoring system that emulates helpful co-constructed relations during human tutoring effective? In *International Conference on Artificial Intelligence in Education*, pages 3–12. Springer, 2015.

[4] Vasile Rus, Dan Stefanescu, Nobal Niraula, and Arthur C Graesser. Deeptutor: towards macro-and micro-adaptive conversational intelligent tutoring at scale. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 209–210, 2014.

[5] Matthew Ventura, Maria Chang, Peter Foltz, Nirmal Mukhi, Jessica Yarbro, Anne Pier Salverda, John Behrens, Jae-wook Ahn, Tengfei Ma, Tejas I Dhamecha, et al. Preliminary evaluations of a dialogue-based digital tutor. In *International Conference on Artificial Intelligence in Education*, pages 480–483. Springer, 2018.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Technical report, 2018. `arXiv:1810.04805`.

[7] Direct Transfer of Learned Information Among Neural Networks. *AAAI-91 Proceedings*, pages 584–589, 1991. URL: www.aaai.org.

[8] Lia M Daniels and Okan Bulut. Students' perceived usefulness of computerized percentage-only vs. descriptive score reports: Associations with motivation and grades. *Journal of Computer Assisted Learning*, 36(2):199–208, 2020.

[9] Anna Filighera, Tim Steuer, and Christoph Rensing. Fooling automatic short answer grading systems. In *International Conference on Artificial Intelligence in Education*, pages 177–190. Springer, 2020.

[10] Chul Sung, Tejas Indulal Dhamecha, and Nirmal Mukhi. Improving short answer grading using transformer-based pre-training. In *International Conference on Artificial Intelligence in Education*, pages 469–481. Springer, 2019.

[11] Leon Camus and Anna Filighera. Investigating transformers for automatic short answer grading. *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II*, 12164:43–48, 06 2020.

[12] Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. WT5?! training text-to-text models to explain their predictions, apr 2020. URL: https://arxiv.org/abs/2004.14546, `arXiv:2004.14546`.

[13] Sasi Kiran Gaddipati, Deebul Nair, and Paul G Plöger. Comparative evaluation of pretrained transfer learning models on automatic short answer grading. *arXiv preprint arXiv:2009.01303*, 2020.

[14] Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 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, 2011.

[15] Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. A diagnostic study of explainability techniques for text classification. *arXiv preprint arXiv:2009.13295*, 2020.

[16] Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUS). Technical report. `arXiv:1606.08415v3`.

[17] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. Technical report, 2016. URL: http://arxiv.org/abs/1607.06450, `arXiv:1607.06450`.

[18] Ashish Vaswani, Google Brain, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. Technical report. URL: https://arxiv.org/pdf/1706.03762.pdf, `arXiv:1706.03762v5`.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, and Paul G Allen. RoBERTa: A Robustly Optimized BERT Pretraining Approach. Technical report,

2019. URL: https://github.com/pytorch/fairseq, arXiv:1907.11692v1.

[20] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. arXiv:1910.01108.

[21] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut, and Google Research. ALBERT: A Lite Bert for Self-Supervised Learning of Language Representations. Technical report. arXiv:1909.11942v3.

[22] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL: http://arxiv.org/abs/1503.02531.

[23] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. pages 784–789, 01 2018. doi:10.18653/v1/P18-2124.

[24] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P16-1162, doi:10.18653/v1/P16-1162.

[25] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. Technical report. URL: http://code.google.com/p/cuda-convnet/, arXiv:1312.6034v2.

[26] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning Important Features Through Propagating Activation Differences. Technical report. URL: http://goo.gl/qKb7pL,, arXiv:1704.02685v2.

[27] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[28] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.

[29] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[30] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[31] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017.

[32] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[33] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.