Improving Short Answer Grading Using Large Transformer-Based Pre-training

Evaluation of state of the art language models for short answer grading
Bachelor thesis in Computer Science by Leon Oliver Camus
Date of submission: May 15, 2020

1. Review: Prof. Dr. rer. nat. Karsten Weihe
2. Review: Julian Prommer, M.Sc.
Darmstadt – D 17

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Computer Science
Department
Algorithmik

Erklärung zur Abschlussarbeit gemäß §23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, , die vorliegende Bachelorarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, den 15. Mai 2020

Leon Oliver Camus

Contents

1 Introduction

In the year 2015 about 28445 subjects where visited by students of the TU Darmstadt [11]. Most of the exercises done in either practices, exercises or exams, of those students are unstructured text based answers. Evaluating those answers proofed to be hard for classical natural language processing based approaches [32] and are work intensive to do by hand. In 2017 the transformer [35] was born setting the new state of the art in translation, outperforming RNN sequence-to-sequence models [36] and the Berkley-Parser [26]. Lately the Transformer was augmented by many teams around the world using new pre-training techniques on large text corpus yielding in even more impressive and larger models like, ELMO [24], BERT [7], GPT-2 [27] and XLM [14].

Prior to this work, other deep learning based approaches have been explored in the context of short answer grading [1, 13, 18, 21, 30, 33]. One of the core constraints of short answer grading remained the limited availability of labeled domain-relevant training data. This issue was mitigated by transfer learning from models pre-trained using unsupervised pre-training tasks, as shown by Sung et al. [33].

The idea of transfer learning is to adapt capabilities obtained by the previous task, to aid the current task. In case of *BERT* [7], this is the task of masked token prediction. Masked token prediction describes the process of randomly taking tokens of a sentence and replacing them with a special token, symbolizing the network that the tokens should be predicted. Next the network should classify the token, reproducing the underlying sentence. This task allows the model to capture general information about sentence structures in the target language. Furthermore, the language model can learn contextualized representations of words, hence knowledge about that word. Since this task does not require any supervision, one could use any corpus to train their model on, such as Wikipedia, Books, news articles or scraped web pages[1]. In this work, I experiment with fine-tuning the most common transformer models and explore the following questions:

[1] `https://commoncrawl.org/`

- Does the size of the Transformer matter for short answer grading?
- How well do multilingual Transformers perform?
- How well do multilingual Transformers generalize to another language?
- Are there better pre-training tasks for short answer grading?
- Does knowledge distillation work for short answer grading?
- Can the performance of the model be improved by multiple reference answers?

1.1 Overview

This work starts with presenting previous related work (Chaper 2). After this brief introduction into I will explain some natural language processing basics (Section 2.1) and fundamental model structures used in natural language processing context (Sections 2.2, 2.3). Yielding into an explanation of the transformer architecture and the underlying attention mechanism (Section 2.4). With the knowledge gathered on the statistical models used in this work, I introduce the recent trend to large scale unsupervised language model pre-training (Section 2.5).

This the insights gathered from this I will explain the experiments I run (Chapter 3). After reviewing the datasets, I used in this study (Section 3.1.1), I will continue showing off the models I used (Section 3.1.2) and elaborate on the resources and parameters I used (Section 3.1.3). After that I will present, the experiments result and analyse them (Section 3.1.4). In the next Section (3.2), I will focus more on the details of using the same model in multiple contexts, presenting ways of turning the three way classification into a two way classification (Section 3.2.1) and ways of dealing with multiple reference answers (Section 3.2.2). In the last Section of this Chapter, I will present a platform for experimenting with the models (Section 3.3) and present qualitative results of a user study I did.

In the final Chapter (Chapter 4), I will explain the usages of such a model (Section 4.1) and present ways to extent this work (Section 4.2).

2 Related Work

The field of short answer grading can mainly be categorized into two classes of approaches. The first ones represent the traditional approaches, based on handcrafted features [19, 20] and the second ones are deep learning based approaches.

Building up on the advances in deep learning research several researchers tried to leverage innovative models for short answer grading. Using deep learning Sultan et al. [32] trained a language model on a vast corpus and used only the sum of its embeddings to calculate the similarity between the student answer and the reference answer. Since embeddings do not contain any contextualized information, nor incorporate sentence structure, information was lost during classification. To extend upon this issue Mueller and Thyagarajan [21] proposed a network consisting of two long-term short-term memory networks for short answer grading. One fundamental issue they encountered and needed to overcome, was the huge amount of training data needed to train artificial neural networks. They used word replacement techniques to artificially extend the training data. The absence of substantial amounts of training data remained a bottleneck for many natural language processing areas. With Vaswani et al. [34] the Transformer architecture was introduced. As a large non-recurrent model it needs even more training data than its predecessors (LSTM and RNN based approaches). Besides the trend to adopt larger models, another promising branch in deep learning emerged, unsupervised pre-training. With pre-trained models, like *InferSent* [4], *ELMo* [25], *BERT* [7], *RoBERTa* [16], *XLM* [14] and *AlBERT* [15], it is possible to finetune a large powerful language model on more limited amounts of data. Recently Sung et al [33] showed that using mentioned techniques, one is able to train a effective and more robust model capable of outperforming previous approaches by about twelve percent. In this study, I aim to extend upon the insights provided by Sung et al [33].

This work uses a special kind deep learning model, called the transformer. To better understand the inner workings and differences of the model, I will give a brief explanation of some NLP and Deep Learning basics needed in this context. After explaining those, I

will give a transformer specific explanation. I assume knowledge of basic machine learning with gradient decent models and a basic understanding of the layer-wise structure of deep learning models.

2.1 Natural Language Processing

Natural language processing (NLP) describes the discipline of natural language understanding by computers. It started in the 1950s as a field closely related to linguistics. With its maturation it incorporated more disciplines such as artificial intelligence, lexicography and statistics.

2.1.1 Tokenization

The process of Tokenization is referred to the splitting of character sequences into tokens. The Entity doing the Tokenization is referred to as Tokenizer. The Tokenizer uses the a given rule-set to split a corpus apart. The unique summary of the created tokens is the vocabulary of the corpus. There a several rule-sets in use today. Every rule-set follows a different idea, for example the Sonority Sequencing Principle, using syllable breaks, tokenizing ”justification” as ”jus”, ”ti”, ”fi”, ”ca”, ”tion”, while the Stanford Tokenizer would not split ”justification” but will split ”can’t” into ”ca”, ”n’t” as it consists of to words.

2.1.2 Word Embeddings

In machine learning, as it depends on statistic models, everything has to be represented by numbers, therefore the words used in language modelling have to be represented as numbers or vectors. An Embedding is a learned vector representation of a word. There are several word vectors like GloVe [23] or Word2Vec and also learned ones loke ELMo [24] or BERT [7]. Word vectors allow vector arithmetic on word level.

$$BERLIN - GERMANY + ENGLAND = LONDON$$

$$PRINCE - MAN + WOMAN = PRINCESS$$

2.2 Recurrent Neural Network

A recurrent neural network (RNN) is a special category of neural networks incorporating a self directed node. This way the neural network is able to compute arbitrarily long input sequences. This is realised using the output of the node of the previous iteration as input for the next iteration as illustrated in figure 2.2. After unrolling this recursion, the model is fully differentiable, assuming a start value for the first recursion.

Assuming a sequence $x = (x_0, x_1, ..., x_t)$, we are able to iteratively compute the output of each stage of the model. Sharing the weight in each iteration, the gradient is able to flow freely from every output to its input and its predecessors, since every output can be written as in equation 2.1. The function A_a resembles the transformation of the previous hidden state and the current input to the next hidden state. The function A_h, acts in similar fashion as A_a, but it outputs the next value h.

$$\begin{aligned} a^{(0)} &= A_a(0, x_0) \\ a^{(n)} &= A_a(a^{(n-1)}, x_n) \\ h_n &= A_h(a^{(n-1)}, x_n) \end{aligned} \tag{2.1}$$

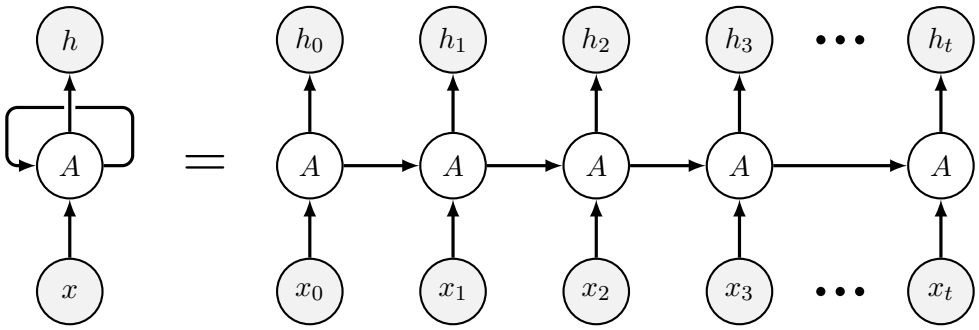

Figure 2.1: A schematic of a rnn cell

2.3 Long Short-Therm Memory

Long short-term memory units are an extended RNN-Unit. It consists of an input gate i_t, an output gate o_t, a forget gate f_t and like a RNN it holds a hidden state c_t and has an

output h_t.

$$
\begin{aligned}
i_t &= \sigma(w_i(h_{t-1}, x_t)^T + b_i) \\
o_t &= \sigma(w_o(h_{t-1}, x_t)^T + b_o) \\
f_t &= \sigma(w_f(h_{t-1}, x_t)^T + b_f)
\end{aligned} \tag{2.2}
$$

$$
c_t = f_t c_{t-1} + i_t x_t \tag{2.3}
$$

$$
h_t = c_t o_t \tag{2.4}
$$

The input and the last output are fed into the gates 2.2. The next hidden state is calculated by masking the last hidden state by multiplying it with the output of the forget gate and adding the, by the input gate, masked input to the LSTM 2.3. To calculate the new output, the hidden state is masked by the output gate 2.4. The LSTM is an improvement to a regular RNN, due to issues of the RNN to hold its information for a longer period steps, due to the explicit gate architecture [10].

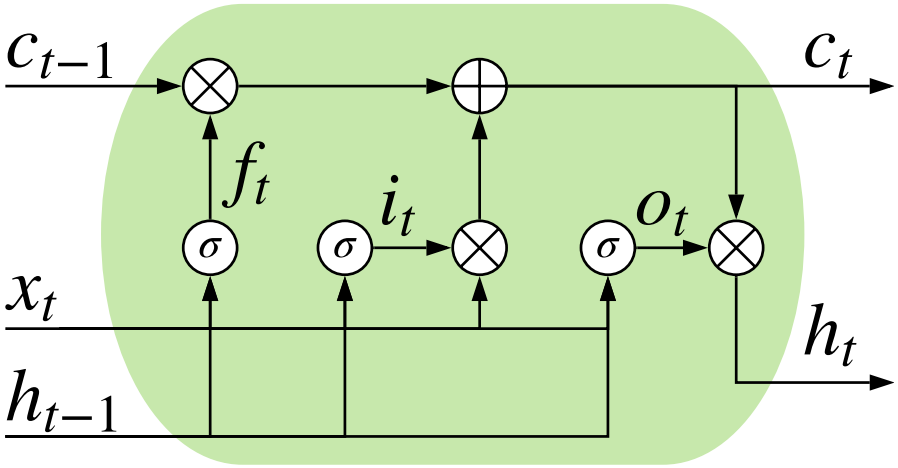

Figure 2.2: A schematic of a lstm unit

2.4 Transformer Model

In this section I want to explain some details about the architectures and pre-training techniques used by the models this work elaborates on. We will first explain some architectural specialities of the models and subsequently explain the pre-training techniques used to incorporate general knowledge into the models.

The transformer model introduced by the paper ”Attention is All You need” [35], consists of two types of modules stacked on top of each other. One builds out of Multiple *Scaled Dot-Product Attention*s 2.5 and the other consisted out of two linear projections with a relu activation function in between 2.6.

$$Norm(x + MultiHeadAttn(x, x, x)) \tag{2.5}$$

$$Norm(x + ReLU(xW_1 + b_1)W_2 + b_2) \tag{2.6}$$

2.4.1 Attention

Attention was used since a long time in language models [3, 36], the *Transformer* also uses an attention mechanism, called the ”Scaled Dot-Product Attention” [35].

Scaled Dot-Product Attention

$$Attention(Q, L, V, M) = softmax(\frac{QK^T}{\sqrt{dim(K)}}M)V \tag{2.7}$$

$$Attention(Q, L, V) = Attention(Q, L, V, 1) \tag{2.8}$$

The ”Scaled Dot-Product Attention” consists of three inputs, the queries Q, the keys K and the values V and one attended output 2.8. If the input is for example padded, a mask M can be added 2.7 [35]. The Scaling factor is introduced to prevent the gradients from vanishing due to the softmax decreasing difference if QK^T runs into larger values, the scaling counteracts this effect [35].

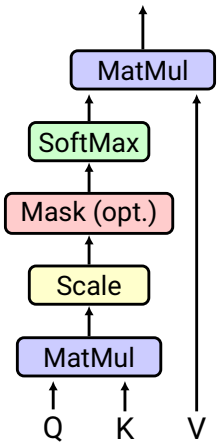

Figure 2.3: Scaled Dot-Product Attention[35]

Multi-Head Attention

$$head_i = Attention(QW_i^Q, LW_i^L, VW_i^V) \tag{2.9}$$

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_n)W^O \tag{2.10}$$

With ”Multi-Head Attention” the queries Q, the keys K and the values V are linearly projected and attended 2.9 and the results are concatenated and again linearly projected 2.10 [35].

Positional Encoding

Because the *Transformer* does not contain any recurrence nor convolution, it needs a positional encoding to make use of the order of the sequence. This is why ”positional encodings” are applied. The positional encodings consist of sine and cosine functions of different frequencies. ”Attention Is All You Need” suggests:

$$PE(pos, 2i) = sin(pos/10000^{2i/d_{model}})$$

$$PE(pos, 2i + 1) = cos(pos/10000^{2i/d_{model}})$$

2.5 Unsupervised Language Model Pre-training

In this section I will elaborate on unsupervised pre-training tasks, leveraging unlabeled raw corpus. To date, there are several pre-training techniques used in the context of Transformers, the most popular and best known is *Masked Language Modeling (MLM)* introduced by Devlin et al. [7]. It was one of the pre-training techniques used to create *BERT*. In *MLM* we randomly replace a token in the input sentence with a special token. The model is then trained to predict the masked token using its context, like in a Cloze test (used, among other tests, to evaluate ones skill in communicating in a foreign language). Devlin et al. [7] also utilized another pre-training task namely *Next Sentence Prediction (NSP)*. In this task we hand the model a sentence, its following sentence and a random sentence picked from the dataset. Next we try to predict which of the later sentences followed the first one. This task was dropped in future work (*RoBERTa*) [16], since it seemed to easy for the model to predict, hence there was not much to learn. Another popular pre-training task was introduced by Yang et al. [39] with their *XLNet*. They used *Permutation Language Modeling (PLM)* to train their model. In this task they randomize the order of a sentence and the model is then trained to predict the proper order of words. In Figure 2.5 I provide an overview over common transformer based models and their pre-training tasks.

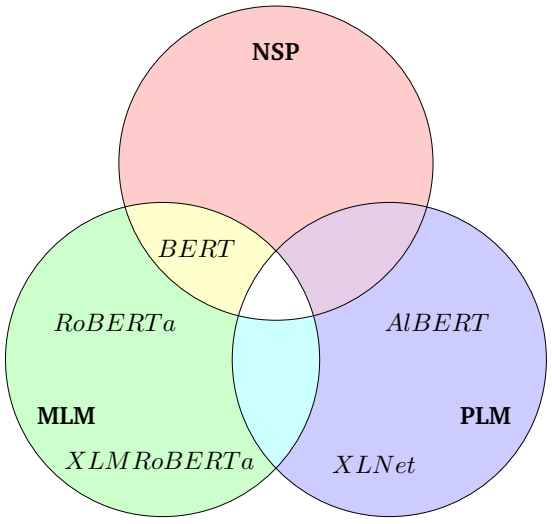

Figure 2.4: Transformer by pre-training task

3 Experiments

In this chapter I want to elaborate on the different types of experiments I did. Firstly, I will explain the technical innovations and deep learning experiments I did. In this section I also evaluate the model on its technical features. After that, I will give an overview over evaluation strategies for multiple reference answers and binary evaluation. Finally, I will show the platform I developed for user interaction and evaluate feedback a user study I did.

3.1 Learning

In this section I will elaborate on the deep learning specific parts of this work. I will provide insights on the datasets and the models used, explain the training process and evaluate the technical questions provided in Section 1.

3.1.1 Datasets

In this work I used two datasets. The SemEval-2013 [8] and the MNLI [37] dataset. I will give a brief explanation on both.

SemEval-2013.

I evaluate my proposed approach on the SemEval-2013 [8] dataset. The dataset consists of questions, reference answers, student answers and three-way labels, represenenting the correct, incorrect and contradictory class (Distribution shown in Table 3.1.1). It is made up out of two data sources, the *SciEntsBank* and the *Beetle* dataset. Each student response is manually annotated by three experts. A majority vote was used to obtain the ground

Table 3.1: The sample size class distribution of the SemEval 2013 dataset.

Dataset	Class	Test Dataset			
		unseen answers	unseen questions	unseen domains	train
sciEntsBank	correct	581	982	417	5134
	incorrect	832	1440	2228	7718
	contradictory	989	2210	1917	9315
	all	**2402**	**4632**	**4562**	**22167**
beetle	correct	523	918	-	4635
	incorrect	583	1072	-	5256
	contradictory	756	1909	-	7307
	all	**1862**	**3899**	**-**	**17198**

truth labels. Furthermore, we use a neural machine translation approach to translate the dataset to the german language and also evaluate on it respectively. Table 3.1 shows the distribution of classes in the train and test data-set. Since the classes are imbalanced, like Sung et al. [33], I will primarily evaluate on the macro-averaged-F1 score, but also report weighted-average-F1 score and accuracy.

MNLI.

I perform transfer learning from a model previously fine-tuned on the MNLI [37] dataset. "The Multi-Genre Natural Language Inference (MultiNLI) corpus is a crowd-sourced collection of 433k sentence pairs annotated with textual entailment information. The corpus is modeled on the SNLI corpus, but differs in that covers a range of genres of spoken and written text, and supports a distinctive cross-genre generalization evaluation."[1] I use the MNLI dataset since the task I want to teach the model is strongly correlated with the MNLI task.

3.1.2 Models

For training and later comparison I utilize the following models:

[1] https://www.nyu.edu/projects/bowman/multinli/

- **BERT$_{\text{LARGE}}$** [7], a larger version of the model use by Sung et al. [33]. In this work we employ the uncased model with whole word masking.

- **BERT$_{\text{DISTILL}}$** [31], a model trained with knowledge distillation from a pre-trained large *BERT* [7] checkpoint. The model with its 66M parameters is about 40% smaller than the base model, with 110M parameters. This reduces the memory and computation needs substantially.

- **RoBERTa$_{\text{BASE}}$** [16], a model similar to the base *BERT* model [7], but with more extensive pre-training on a vaster corpus.

- **RoBERTa$_{\text{LARGE}}$** [16], containing 355M parameters instead of the 125M used by the BASE version.

- **RoBERTa$_{\text{LARGE,MNLI}}$** [16], the same model as the large *RoBERTa* model, but previously fine-tuned on the MNLI [37] dataset.

- **RoBERTa$_{\text{DISTILL}}$** [31], similar to the distilled version of *BERT*, but distilled from the large *RoBERTa* model.

- **XLMRoBERTa$_{\text{BASE}}$** [5], implementing the same architecture as the base *RoBERTa* model, it is trained on 100 different languages using masked language modeling.

- **AlBERT$_{\text{BASE}}$** [15], architecturally similar to the base version of *BERT*, but with weight sharing of all layers. This reduces the memory usage of the model, increases training speed, but leaves computational needs untouched.

- **AlBERT$_{\text{LARGE}}$** [15], architecturally similar to the large version of *BERT*, but with weight sharing of all layers.

3.1.3 Fine-Tuning Setup

For fine tuning I add a classification layer on top of every model. I use the AdamW [17] optimizer, with a learning rate of 2e-5 and a linear learning rate schedule with warm up. For large transformers we extend the number of epochs to 24, but I also observe notable results with 12 epochs or less. I train using a single NVIDIA 2080ti GPU (11GB) with a batch size of 16, utilizing gradient accumulation. Larger batches did not seem to

Table 3.2: Results on the SciEntsBank Dataset of SemEval 2013. Macro-average-F1 (M-F1) is reported in percentage.

	Unseen answer	Unseen question	Unseen domain
	M-F1	M-F1	M-F1
Sung et al. [33]	72.0	57.5	57.9
RoBERTa$_{\text{DISTILL}}$	73.2	55.2	55.6
RoBERTa$_{\text{BASE}}$	73.2	61.7	62.5
RoBERTa$_{\text{LARGE}}$	75.5	62.7	65.6

improve the results. To fit large transformers into the GPU memory I use a combination of gradient accumulation and mixed precision with 16 bit floating point numbers, provided by NVIDIAs apex library[2]. We implement our experiments using huggingfaces transformer library [38]. To ensure comparability, all of the presented models where trained with the same code, setup and hyper parameters.

3.1.4 Results and Analysis

Does the size of the Transformer matter for short answer grading?

Large models demonstrate a significant improvement compared to Base models (see Table 3.2). The improvement arises most likely due to the increased capacity of the model, as more parameters allow the model to retain more information of the pre-training data.

How well do multilingual Transformers perform?

The *XLM* [14] based models do not perform well in this study. The *RoBERTa* based models (*XLMRoBERTa*) seem to generalize better than their predecessors (see Table 3.6). *XLMRoBERTa* performs similarly to the base *RoBERTa* model, falling behind in the unseen questions and unseen domains category (See Table 3.3). Subsequent investigations could include fine-tuning the large variant on MNLI and SciEntsBank. Due to GPU memory constraints, we were not capable to train the large variant of this model.

[2] `https://github.com/NVIDIA/apex`

Table 3.3: Results on the SciEntsBank Dataset of SemEval 2013. Macro-average-F1 (M-F1) is reported in percentage.

	Unseen answer	Unseen question	Unseen domain
	M-F1	M-F1	M-F1
Sung et al. [33]	72.0	57.5	57.9
RoBERTa$_{\text{BASE}}$	73.2	61.7	62.5
XLMRoBERTa$_{\text{BASE}}$	73.8	57.9	54.4

Table 3.4: Results on the SciEntsBank Dataset of SemEval 2013 (English and German), of a mono lingual and a multi lingual model. Macro-average-F1 (M-F1) is reported in percentage.

		English			German		
	Languages	UA	UQ	UD	UA	UQ	UD
	Trained	M-F1	M-F1	M-F1	M-F1	M-F1	M-F1
RoBERTa$_{\text{LARGE}}$	en	75.5	62.7	65.6	40.4	34.7	48.2
RoBERTa$_{\text{LARGE}}$	de	19.4	21.5	19.7	19.4	21.5	19.7
RoBERTa$_{\text{LARGE}}$	en,de	74.9	61.9	63.3	72.3	57.3	56.3
XLMRoBERTa$_{\text{BASE}}$	en	73.8	57.9	54.4	60.6	48.3	49.5
XLMRoBERTa$_{\text{BASE}}$	de	67.4	51.9	51.9	71.7	55.6	49.7
XLMRoBERTa$_{\text{BASE}}$	en,de	72.4	57.5	48.1	71.3	54.6	46.5

How well do multilingual Transformers generalize to another language?

The models with multilingual pre-training show stronger generalization across languages than their English counterparts. In Table 3.4 we are able to observe that the score of the multilingual model increases across languages it was never fine-tuned on, while the monolingual model does not generalize.

Are there better pre-training tasks for short answer grading?

Transfer learning a model from MNLI yields a significant improvement over the same version of the model not fine-tuned on MNLI (See Table 3.5). This indicates, that the

Table 3.5: Results on the SciEntsBank Dataset of SemEval 2013 (English and German). Macro-average-F1 (M-F1) is reported in percentage.

		English			German		
	Languages	UA	UQ	UD	UA	UQ	UD
	Trained	M-F1	M-F1	M-F1	M-F1	M-F1	M-F1
RoBERTa$_{\text{LARGE}}$	en	75.5	62.7	65.6	40.4	34.7	48.2
RoBERTa$_{\text{LARGE}}$	de	19.4	21.5	19.7	19.4	21.5	19.7
RoBERTa$_{\text{LARGE}}$	en,de	74.9	61.9	63.3	72.3	57.3	56.3
RoBERTa$_{\text{LARGE,MNLI}}$	en	78.3	65.7	70.8	49.3	42.5	51.9
RoBERTa$_{\text{LARGE,MNLI}}$	de	59.1	51.5	66.8	74.0	59.0	57.2
RoBERTa$_{\text{LARGE,MNLI}}$	en,de	79.1	65.3	69.1	75.0	58.4	59.2

models acquire important skills from the MNLI dataset, which it does not learn by training only on the SemEval dataset.

Those skills improve the models abilities to generalise to a separate domain. The models capabilities on the german version of the dataset are also increased, despite the usage of a monolingual model. Furthermore training a model, which has never seen the German language (*RoBERTa* large, MNLI), trained German and English version dataset seem to generalize better.The reason for this behavior should be further investigated.

Does knowledge distillation work for short answer grading?

The usage of models pre-trained with knowledge distillation yields a slightly lower score. However, since the model is 40% smaller, a maximum decrease in performance of about 2% to the previous state of the art (see Table 3.2) may be acceptable for scenarios where computational resources are limited.

3.2 Inference

In this section I will show different ways of evaluating the models. Firstly I will show a formulation for projecting a model trained in three way classification for two way

Table 3.6: Results on the SciEntsBank Dataset of SemEval 2013. Accuracy (Acc), macro-average-F1 (M-F1), and weighted-average-F1 (W-F1) are reported in percentage.

	Languages	English									German								
		Unseen answer			Unseen question			Unseen domain			Unseen answer			Unseen question			Unseen domain		
	Trained	Acc	M-F1	W-F1	Acc	M-F1	W-F1	Acc	M-F1	W-F1	Acc	M-F1	W-F1	Acc	M-F1	W-F1	Acc	M-F1	W-F1
Baseline [8]	en	55.6	40.5	52.3	54.0	39.0	52.0	57.7	41.6	55.4	-	-	-	-	-	-	-	-	-
ETS [9]	en	72.0	64.7	70.8	58.3	39.3	53.7	54.3	33.3	46.1	-	-	-	-	-	-	-	-	-
SOFTCAR [12]	en	65.9	55.5	64.7	65.2	46.9	63.4	63.7	48.6	62.0	-	-	-	-	-	-	-	-	-
MEAD [29]	en	-	42.9	55.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Graph [29]	en	-	43.8	56.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Sultan et al. [32]	en	60.4	44.4	57.0	64.3	45.5	61.5	62.7	45.2	60.3	-	-	-	-	-	-	-	-	-
Saha et al. [30]	en	71.8	66.6	71.4	61.4	49.1	62.8	63.2	47.9	61.2	-	-	-	-	-	-	-	-	-
Marvaniya et al. [18]	en	-	63.6	71.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Sung et al. [33]	en	75.9	72.0	75.8	65.3	57.5	64.8	63.8	57.9	63.4	-	-	-	-	-	-	-	-	-
BERT$_{\text{DISTILL}}$	en	69.2	67.2	69.2	56.6	54.7	56.6	61.4	49.7	61.4	38.8	26.2	38.8	33.7	21.5	33.7	51.7	29.5	51.7
BERT$_{\text{BASE}}$	en	72.8	70.6	72.8	57.3	56.0	57.3	63.4	54.6	63.4	45.0	37.0	45.0	40.5	32.9	40.5	59.0	41.7	59.0
BERT$_{\text{LARGE}}$	en	75.8	75.0	75.8	63.4	62.4	63.4	67.7	62.8	67.7	50.2	40.5	50.2	43.7	35.5	43.7	57.7	42.2	57.7
RoBERTa$_{\text{DISTILL}}$	en	74.8	73.2	74.8	56.9	55.2	56.9	65.1	55.6	65.1	48.0	40.4	48.0	44.1	36.7	44.1	58.2	40.2	58.2
RoBERTa$_{\text{BASE}}$	en	74.5	73.2	74.5	63.2	61.7	63.2	65.3	62.5	65.3	47.8	38.1	47.8	43.5	35.1	43.5	60.6	43.2	60.6
RoBERTa$_{\text{LARGE}}$	en	76.7	75.5	76.7	64.1	62.7	64.1	66.8	65.6	66.8	48.8	40.4	48.8	42.7	34.7	42.7	61.5	48.2	61.5
RoBERTa$_{\text{LARGE}}$	de	41.2	19.4	41.2	47.7	21.5	47.7	42.0	19.7	42.0	41.2	19.4	41.2	47.7	21.5	47.7	42.0	19.7	42.0
RoBERTa$_{\text{LARGE}}$	en,de	76.1	74.9	76.1	63.0	61.9	63.0	65.6	63.3	65.6	73.9	72.3	73.9	58.9	57.3	58.9	61.9	56.3	61.9
RoBERTa$_{\text{LARGE,MNLI}}$	en	78.8	78.3	78.8	**66.4**	**65.7**	**66.4**	**71.8**	**70.8**	**71.8**	52.6	49.3	52.6	46.1	42.5	46.1	60.1	51.9	60.1
RoBERTa$_{\text{LARGE,MNLI}}$	de	62.6	59.1	62.6	55.1	51.5	55.1	66.5	66.8	66.5	74.9	74.0	74.9	**60.8**	**59.0**	**60.8**	62.5	57.2	62.5
RoBERTa$_{\text{LARGE,MNLI}}$	en,de	**79.7**	**79.1**	**79.7**	66.3	65.3	66.3	69.4	69.1	69.4	**76.0**	**75.0**	**76.0**	59.6	58.4	59.6	**64.9**	**59.2**	**64.9**
AlBERT$_{\text{BASE}}$	en	72.6	71.4	72.6	57.6	55.2	57.6	60.1	52.3	60.1	37.0	31.5	37.0	30.0	24.9	30.0	40.6	31.8	40.6
AlBERT$_{\text{LARGE}}$	en	71.3	70.1	71.3	58.1	56.8	58.1	65.3	60.7	65.3	45.0	42.1	45.0	38.7	34.9	38.7	58.1	46.0	58.1
XLM$_{\text{MLM-TLM-XNLI}}$	en	72.6	71.2	72.6	57.6	55.5	57.6	56.3	44.8	56.3	48.0	47.4	48.0	43.7	43.3	43.7	47.7	41.5	47.7
XLM$_{\text{MLM-TLM-XNLI}}$	de	57.0	54.8	57.0	43.7	41.9	43.7	56.4	41.2	56.4	68.8	66.5	68.8	54.0	51.6	54.0	55.8	44.8	55.8
XLM$_{\text{MLM-TLM-XNLI}}$	en,de	64.8	62.2	64.8	52.1	49.2	52.1	48.6	35.7	48.6	63.8	61.2	63.8	51.8	49.7	51.8	48.1	33.8	48.1
XLMRoBERTa$_{\text{BASE}}$	en	75.4	73.8	75.4	59.9	57.9	59.9	62.6	54.4	62.6	64.2	60.6	64.2	52.7	48.3	52.7	60.1	49.5	60.1
XLMRoBERTa$_{\text{BASE}}$	de	69.0	67.4	69.0	53.6	51.9	53.6	62.3	51.9	62.3	73.4	71.7	73.4	56.9	55.6	56.9	60.8	49.7	60.8
XLMRoBERTa$_{\text{BASE}}$	en,de	74.1	72.4	74.1	59.1	57.5	59.1	60.1	48.1	60.1	73.1	71.3	73.1	56.1	54.6	56.1	59.4	46.5	59.4

regression.

3.2.1 Projection

For the 2-way I combine the value of contradictory and incorrect 3.2, since contradictory also means false in this context. For regression I multiply the incorrect class by minus one and add it to the correct class 3.3.

$$score_{3way} = \begin{pmatrix} correct_{3way} \\ incorrect_{3way} \\ contradictory_{3way} \end{pmatrix} = score_0 + score_1 + ... + score_n \tag{3.1}$$

$$score_{2way} = \begin{pmatrix} correct_{2way} \\ incorrect_{2way} \end{pmatrix} = \begin{pmatrix} correct_{3way} \\ incorrect_{3way} + contradictory_{3way} \end{pmatrix} \tag{3.2}$$

$$score_{reg} = correct_{2way} - incorrect_{2way} \tag{3.3}$$

Using this equation I was able to create ROC-Curves for the best performing model **RoBERTa**$_{\text{LARGE,MNLI}}$, as shown in figure 3.2.1

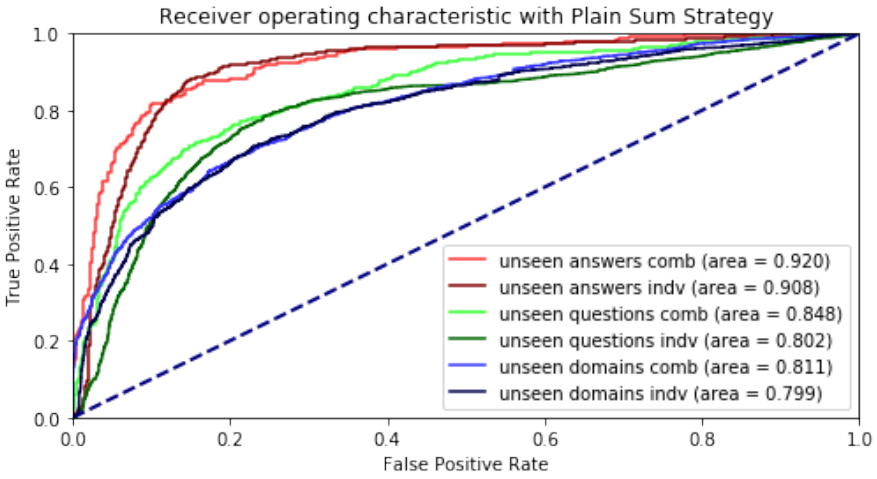

Figure 3.1: The ROC Curve of *RoBERTa* Large (ind - without combination of the scores, comb - with combination of the scores)

3.2.2 Multiple Reference Answers

In real-life applications we have different amounts of reference answers, combining classifications of different models and evaluating with different amounts of reference answers is necessary to get a glimpse on the performance of those models in production. The evaluation uses several ways of combining the outputs of the models into a 3-way, 2-way and regression like value. For creating the scores I add up the vectors outputted by the models 3.1. Normalisation of the scores did not improve the accuracy. Using this I am able to create ROC curves for different amounts of reference answers (Figure 3.2.2 or Table 3.2.2). I also used this technique to evaluate the model by its scores, grouped by

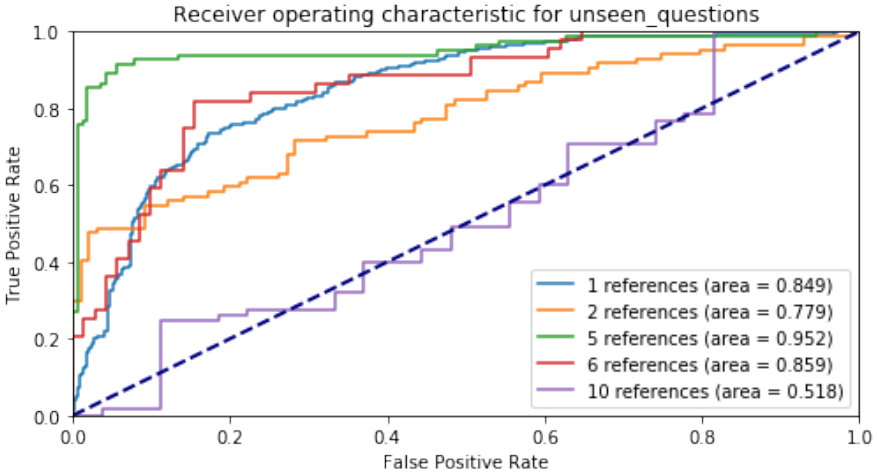

Figure 3.2: The ROC Curve of *RoBERTa* Large grouped by reference count

the amount of reference answers (Figure 3.2.2 or Table 3.2.2). I will now evaluate the insights gathered by this evaluation.

Can the performance of the model be improved by multiple reference answers?

More reference answers seem to improve the overall performance. In my studies I observe a peak performance at three to five reference answers. This is a very rough estimate, since with more references the support of those numbers is decreasing and also the difficulty of the answers may not be the same. But nevertheless I am able to observe a trend that more reference answers up to around five are increasing precision.

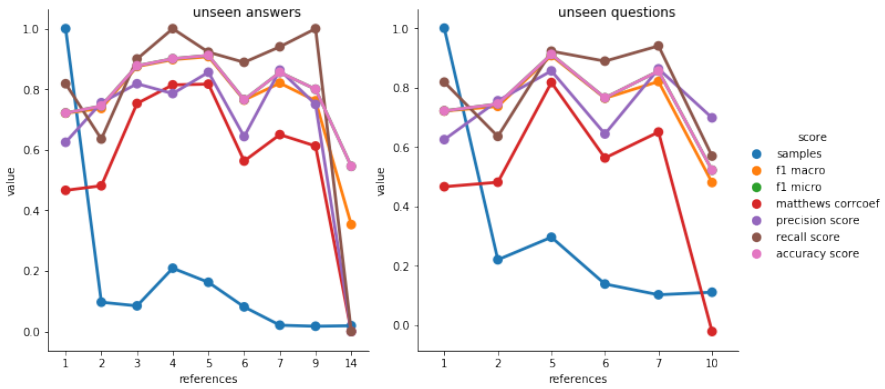

Figure 3.3: Evaluation scores of the *RoBERTa* Model with the relative support, grouped by reference count

Table 3.7: Evaluation on Semeval Unseen Answers

References	Samples	F1 Macro	F1 Micro	Accuracy
1	579	74.87	75.54	75.54
2	56	75.16	76.57	76.57
3	**49**	**93.61**	**93.88**	**93.88**
4	121	92.99	93.39	93.39
5	94	91.90	92.35	92.35
6	47	80.73	81.48	81.48
7	12	83.85	86.60	86.60
9	10	76.19	80.00	80.00
14	11	35.29	54.55	54.55

Table 3.8: Evaluation on Semeval Unseen Questions

References	Samples	F1 Macro	F1 Micro	Accuracy
1	831	74.87	75.54	75.54
2	183	75.16	76.57	76.57
5	**246**	**91.90**	**92.35**	**92.35**
6	115	80.73	81.48	81.48
7	85	83.85	86.60	86.60
10	92	42.96	45.65	45.65

3.3 Human Evaluation

Using *Vue.js*, *Boostrap* and *Flask* a interactive frontend was build to interact with the given models. The technologies where chosen cause of easy integration and already established knowledge.

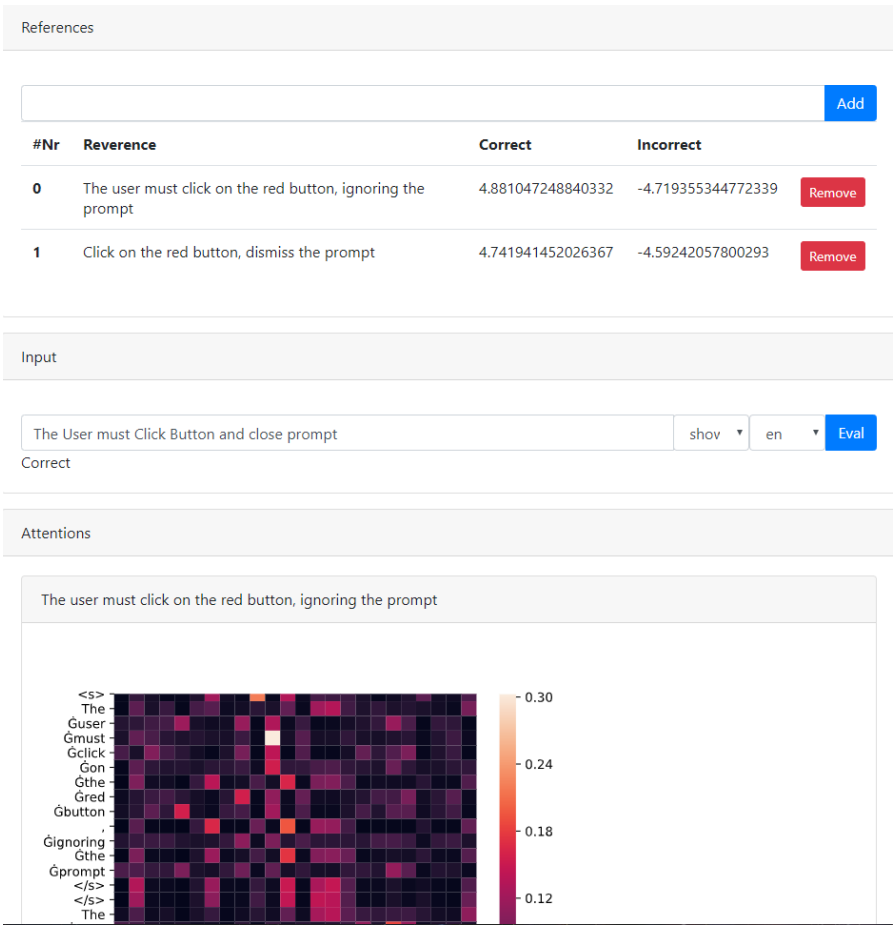

Figure 3.4: The frontend build with Bootstrap, Vue.js

Experimenting with different answers yields disturbing results. First off printing the attention heads gives us a hint what the model sees as relevant information and which

words are seen together in a context. We are able to see that the attention head attends

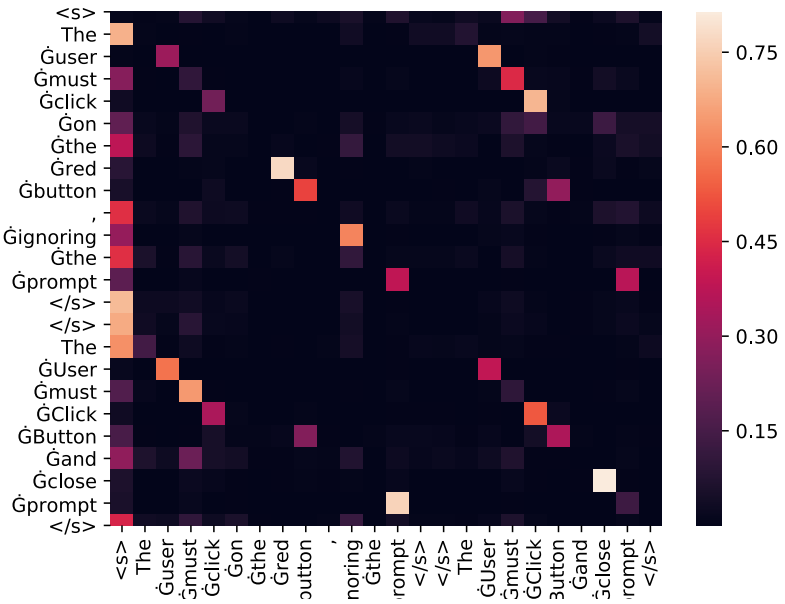

Figure 3.5: Attention matrix of head ten in layer one

nearly every token to the initial space token, furthermore it seems to be only looking for similar words to attend. With this information we are able to take a guess and try to throw away all semantic information from our given answer. So we try the answer: ”Red button” resulting in a categorisation of Correct. Also more domain specific questions are difficult to grade by the model, most likely due to the special implications of some domain specific words.

3.3.1 User Study

Since to the end of this work, the Covid-19 crisis spread. This study had to be moved into the digital area. With this I lost the ability to do a larger case study. Because of this I will only present qualitative results and give insights on the core extensions the users request. This study had 7 participants. I presented the prototype to three scientific assistants, two lecturers and two students. The experimentees where selected from the fields: FB20,

FB4, FB18, FB13, FB16. Everyone has described the platform to be intuitive and easy to use. The provided examples worked well, for most users. The users where given the opportunity to design own question answer pairs. I presented four points of extension the experimentees should put in one of four categories:

- I would enjoy having this feature (Original: Das würde mich sehr freuen).
- I need this feature (Original: Das setze ich voraus).
- I do not care about this feature (Original: Das ist mir egal).
- I oppose this feature (Original: Das nehme ich gerade noch hin).
- I strongly oppose this feature (Original: Das würde mich sehr stören).

How would you react to provide grading to the related reference answers? (Was würden Sie sagen, wenn Sie die Referenzantworten mit Punkten belegen könnten um die Güte der Referenzantwort zu beschreiben?)

With this question I can observe a strong consent to needing this feature to use the automatic short answer grading in their subjects. In this question three people voted for “I would enjoy having this feature”, three people for “I need this feature” and one voted for “I oppose this feature”.

How would you react if we coloured similar words of reference and provided answers similar? (Was würden Sie sagen, wenn gleiche Worte in der Referenzantwort und Referenzlösung farblich markiert werden?)

Digging through the answers of this question the consent seems to be “I would enjoy having this feature”, indicating the user do not see this feature as an essential, but beneficial. The answers consist out of four people who ticked “I need this feature”, one experimentee who ticked “I strongly oppose this feature” and four people who ticked “I would enjoy having this feature”.

How would you react to another class, contradictory? (Was würden Sie sagen, wenn es eine weitere Behauptung gäbe: ”gegensätzlich”?)

With one abstention a truce occurred between “I would enjoy having this feature” and “I need this feature”.

How would you react, if you could provide additional context to the model? (Was würden Sie sagen, wenn es eine Möglichkeit gäbe Texte mit Wissensgrundlagen für die maschinellen Lerner zu hinterlegen?)

This question was quite controversial, but some commented they did not understand the scope of the questions, additionally there where two abstentions. The distribution of answers are two who ticked “I would enjoy having this feature”, two who ticked “I need this feature” and one who ticked “I do not care about this feature”.

4 Conclusion

This chapter summarises the results of the previous experiments and presents different ways of extending this work.

4.1 Viability of Usage

In this work I demonstrate that large Transformer-based pre-trained models achieve state of the art results in short answer grading. I am able to show that models trained on the MNLI dataset are capable of transferring knowledge to the task of short answer grading. Moreover, I am able to increase a models overall score, by training it on multiple languages. I show that the skills developed by a model trained on MNLI improve generalization across languages. It is also shown, that cross lingual training improves scores on SemEval2013. I show that knowledge distillation allows for good performance, while keeping computational costs low. This is crucial in evaluating answers from many users, like in online tutoring platforms. The resulting model outperforms every model previous tested. But it still got issues, like the sensibility to ”trigger words” and the low semantic awareness of the model, which make it easily exploitable in an interactive Scenario. It is also difficult to deploy the model onto questions in very specialized domains, since the model is not able to understand the relations and more important the synonyms their related descriptions which results in a poorly performing model in these scenarios. On less specific common sense questions the model performs reasonably well and in combination of a key word matching could be used to give a initial feedback to the user answering the question.
In an E-Learning scenario it would also be beneficial to gain knowledge about why and where the error in a answer arises. But a lack of datasets in the short answer grading domain makes this difficult to accomplish. Also due to the electrical engineering questions present in the *semeval* dataset the model performs reasonably well in this domain, although it is a more specialized one. The models trained on a translated dataset also performed reasonably well given the datasets poor quality. The translated dataset, even

more, degrades the context awareness and the domain specific words. I propose usage of the model in practice exercises. The model could also be employed in semi-automatic grading of examinations, for example in a majority vote with a human corrector, thus leading too reduced correction errors.

4.2 Future Work

Future research should investigate the impact of context on the classification. To be able adapt to a new domain, firstly the model has to understand the domain. The first and most obvious source of context is the question itself. In this work, we only incorporate the reference answer and the student answer into the classification. However, also including the question may help the model grade answers, which were not considered during the reference answer creation. Another source of context derives from the source of the question itself. In a learning context, a question refers to a source of information. Future research could examine how to include this information during classification. The largest source of knowledge and therefore context is the web itself. Using information from the web, a model may be able to generalize even better. Another extension point for future research is the possibility to apply one model for multiple languages. In overcoming the degradation of the dataset due to translation issues, *Mikel Artetxe et al. (2019)* [2] proposed an technique to transfer monolingual representations into other languages, which could be used to improve performance and compare to the translation approach. It is a question of future research to investigate if even larger transformers like the *T5* [28] will keep on improving performance or if we need to develop better architectures or just better pre-training tasks to keep on improving.

Acknowledgements. I would like to thank Prof. Dr. rer. nat. Karsten Weihe, Julian Prommer M.Sc., Anna Filighera M.Sc., the department of didactics and Nena Marie Helfert, for supporting and reviewing this work. I would like to thank my parents Manuela Camus and Jochen Camus for supporting me all my life, always encouraging me in my work and decisions. Finally I want to thank my friends and everyone who cheered me up when I was down, keeping me productive and focused, but also distracted me when I needed distraction.

References

[1] Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. “Automatic text scoring using neural networks”. In: *arXiv preprint arXiv:1606.04289* (2016).

[2] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. “On the cross-lingual transferability of monolingual representations”. In: *arXiv preprint arXiv:1910.11856* (2019).

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).

[4] Alexis Conneau et al. “Supervised learning of universal sentence representations from natural language inference data”. In: *arXiv preprint arXiv:1705.02364* (2017).

[5] Alexis Conneau et al. “Unsupervised cross-lingual representation learning at scale”. In: *arXiv preprint arXiv:1911.02116* (2019).

[6] Alexis Conneau et al. “XNLI: Evaluating Cross-lingual Sentence Representations”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018.

[7] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[8] Myroslava O Dzikovska et al. *Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge*. Tech. rep. NORTH TEXAS STATE UNIV DENTON, 2013.

[9] Michael Heilman and Nitin Madnani. “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)*. 2013, pp. 275–279.

[10] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[11] Dezernat II - Studierendenservice und Hochschulrecht. *Studierendenstatistik TU Darmstadt Sommmersemester 2015*. 8.05.2015.

[12] Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. “SOFTCARDINALITY: Hierarchical text overlap 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)*. 2013, pp. 280–284.

[13] Sachin Kumar, Soumen Chakrabarti, and Shourya Roy. “Earth Mover’s Distance Pooling over Siamese LSTMs for Automatic Short Answer Grading.” In: *IJCAI*. 2017, pp. 2046–2052.

[14] Guillaume Lample and Alexis Conneau. “Cross-lingual language model pretraining”. In: *arXiv preprint arXiv:1901.07291* (2019).

[15] Zhenzhong Lan et al. “Albert: A lite bert for self-supervised learning of language representations”. In: *arXiv preprint arXiv:1909.11942* (2019).

[16] Yinhan Liu et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *arXiv preprint arXiv:1907.11692* (2019).

[17] Ilya Loshchilov and Frank Hutter. “Fixing weight decay regularization in adam”. In: *arXiv preprint arXiv:1711.05101* (2017).

[18] Smit Marvaniya et al. “Creating scoring rubric from representative student answers for improved short answer grading”. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2018, pp. 993–1002.

[19] 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-Volume 1*. Association for Computational Linguistics. 2011, pp. 752–762.

[20] Michael Mohler and Rada Mihalcea. “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)*. 2009, pp. 567–575.

[21] Jonas Mueller and Aditya Thyagarajan. “Siamese recurrent architectures for learning sentence similarity”. In: *thirtieth AAAI conference on artificial intelligence*. 2016.

[22] Nathan Ng et al. “Facebook FAIR’s WMT19 News Translation Task Submission”. In: *arXiv preprint arXiv:1907.06616* (2019).

[23] Jeffrey Pennington, Richard Socher, and Christopher Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[24] Matthew E. Peters et al. “Deep contextualized word representations”. In: *Proc. of NAACL*. 2018.

[25] Matthew E Peters et al. “Deep contextualized word representations”. In: *arXiv preprint arXiv:1802.05365* (2018).

[26] Slav Petrov et al. “Learning accurate, compact, and interpretable tree annotation”. In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. 2006, pp. 433–440.

[27] Alec Radford et al. “Language Models are Unsupervised Multitask Learners”. In: (2019).

[28] Colin Raffel et al. “Exploring the limits of transfer learning with a unified text-to-text transformer”. In: *arXiv preprint arXiv:1910.10683* (2019).

[29] Lakshmi Ramachandran and Peter Foltz. “Generating reference texts for short answer scoring using graph-based summarization”. In: *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*. 2015, pp. 207–212.

[30] Swarnadeep Saha et al. “Sentence level or token level features for automatic short answer grading?: Use both”. In: *International conference on artificial intelligence in education*. Springer. 2018, pp. 503–517.

[31] Victor Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *NeurIPS EMC^2 Workshop*. 2019.

[32] Md Arafat Sultan, Cristobal Salazar, and Tamara Sumner. “Fast and easy short answer grading with high accuracy”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2016, pp. 1070–1075.

[33] Chul Sung, Tejas Indulal Dhamecha, and Nirmal Mukhi. “Improving Short Answer Grading Using Transformer-Based Pre-training”. In: *Artificial Intelligence in Education*. Ed. by Seiji Isotani et al. Cham: Springer International Publishing, 2019, pp. 469–481. isbn: 978-3-030-23204-7.

[34] Ashish Vaswani et al. “Attention is All You Need”. In: 2017. url: `https://arxiv.org/pdf/1706.03762.pdf`.

[35] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[36] Oriol Vinyals et al. “Grammar as a foreign language”. In: *Advances in neural information processing systems*. 2015, pp. 2773–2781.

[37] Adina Williams, Nikita Nangia, and Samuel Bowman. “A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 1112–1122. url: http://aclweb.org/anthology/N18-1101.

[38] Thomas Wolf et al. “HuggingFace’s Transformers: State-of-the-art Natural Language Processing”. In: *ArXiv* abs/1910.03771 (2019).

[39] Zhilin Yang et al. “XLNet: Generalized Autoregressive Pretraining for Language Understanding”. In: *arXiv preprint arXiv:1906.08237* (2019).