Explaining Translationese: why are Neural Classifiers Better and what do they Learn?

Kwabena Amponsah-Kaakyire[*1,2], **Daria Pylypenko**[*1], **Josef van Genabith**[1,2], and **Cristina España-Bonet**[2]
[1]Saarland University, [2]German Research Center for Artificial Intelligence (DFKI)
Saarland Informatics Campus, Saarbrücken, Germany
amponsahkaakyirek@gmail.com
daria.pylypenko@uni-saarland.de
{cristinae, Josef.Van_Genabith}@dfki.de

Abstract

Recent work has shown that neural feature- and representation-learning, e.g. BERT, achieves superior performance over traditional manual feature engineering based approaches, with e.g. SVMs, in translationese classification tasks. Previous research did not show (i) whether the difference is because of the features, the classifiers or both, and (ii) what the neural classifiers actually learn. To address (i), we carefully design experiments that swap features between BERT- and SVM-based classifiers. We show that an SVM fed with BERT representations performs at the level of the best BERT classifiers, while BERT learning and using handcrafted features performs at the level of an SVM using handcrafted features. This shows that the performance differences are due to the features. To address (ii) we use integrated gradients and find that (a) there is indication that information captured by hand-crafted features is only a subset of what BERT learns, and (b) part of BERT's top performance results are due to BERT learning topic differences and spurious correlations with translationese.

1 Introduction

Translationese is a descriptive (non-negative) cover term for the systematic differences between translated and originally authored text in same language (Gellerstam, 1986). Some aspects of translationese such as source interference (Toury, 1980; Teich, 2003) are language dependent, others are presumed universal, e.g. simplification, explicitation, overadherence to target language linguistic norms (Volansky et al., 2015) in the products of translations. While translationese effects can be subtle, especially for professional human translation, corpus-based studies (Baker et al., 1993) and, in particular, machine-learning and classifier based studies (Rabinovich and Wintner, 2015; Volansky et al., 2015; Rubino et al., 2016; Pylypenko et al., 2021) clearly reveal the differences.

While research on translationese is important from a theoretical point of view (translation universals, specific interference), it has a direct impact on machine translation research: (Kurokawa et al., 2009; Stymne, 2017; Toral et al., 2018; Zhang and Toral, 2019; Freitag et al., 2019; Graham et al., 2020; Riley et al., 2020), amongst others, show that translation direction in training and test data impacts on results, that already translated test data are easier to translate than original data, that machine translation and post-editing result in translationese, and that mitigating translationese in MT output can improve results. Translationese impacts cross-lingual applications, e.g. question answering and natural language inference (Singh et al., 2019; Clark et al., 2020; Artetxe et al., 2020).

In this paper we focus on machine-learning-classifier-based research on translationese. Here, typically a classifier is trained to distinguish between original and translated texts (in the same language). Until recently, most of this research (Baroni and Bernardini, 2005; Volansky et al., 2015; Rubino et al., 2016) used manually defined, often linguistically inspired, feature-engineering based sets of features, mostly using support vector machines (SVM). Once a classifier is trained, feature importance and ranking methods are used to reason back to what aspects of the input is responsible for (i.e. explains) the classification (and whether this accords with linguistic theorisation). More recently, a small number of papers explored feature- and representation-learning neural network based approaches to translationese classification (Sominsky and Wintner, 2019). In a systematic study Pylypenko et al. (2021) show that feature- and representation-learning deep neural network-based approaches (in particular BERT-based, but also other neural approaches) to translationese

[*]Equal contribution.

Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pages 281 - 296

classification substantially outperform handcrafted feature-engineering based approaches using SVMs. However, to date, two important questions remain: (i) it is not clear whether the substantial performance differences are due to learned vs. handcrafted features, the classifiers (SVM, the BERT classification head, or full BERT), or the combination of both, and (ii) what the neural feature and representation learning approaches actually learn and how that explains the superior classification. The contributions of our paper are as follows:

1. we address (i) by carefully crossing features and classifiers, feeding BERT-based learned features to feature-engineering models (SVMs), feeding the BERT classification head with hand-crafted features, and by making BERT architectures learn handcrafted features, as well as feeding embeddings of handcrafted features into BERT. Our experiments show that SVMs using BERT-learned features perform on a par with our best BERT-translationese classifiers, while BERT using handcrafted features only performs at the level of feature-engineering-based classifiers. This shows that it is the features and not the classifiers, that lead to the substantial (up to 20% points accuracy absolute) difference in performance.

2. we present the first steps to address (ii) using integrated gradients, an attribution-based approach, on the BERT models trained in various settings. Based on striking similarities in attributions between BERT trained from scratch and BERT pretrained on handcrafted features and fine-tuned on text data, as well as comparable classification accuracies, we find evidence that the hand-crafted features do not bring any additional information over the set learnt by BERT. it is therefore likely that the hand-crafted features are a (possibly partial) subset of the features learnt by BERT. Inspecting the most attributed tokens, we present evidence of 'Clever Hans' behaviour: at least part of the high classification accuracy of BERT is due to names of places and countries, suggesting that part of the classification is topic- and not translationese-based. Moreover, some top features suggest that there may be some punctuation-based spurious correlation in the data.

2 Related Work

Combining learned and hand-crafted features. (Kaas et al., 2020; Prakash and Tayyar Madabushi, 2020; Lim and Tayyar Madabushi, 2020) combine BERT-based and manual features in order to improve accuracy. (Kazameini et al., 2020; Ray and Garain, 2020; Zhang and Yamana, 2020) concatenate BERT pooled output embeddings with handcrafted feature vectors for classification, often using an SVM, where the handcrafted feature vector might be further encoded by a neural network or used as it is. Our work differs in that we do not combine features from both models but swap them in order to decide whether it is the features, the classifiers or the combination that explains the performance difference between neural and feature engineering based models. Additionally, our approach allows us to examine whether or not representation learning learns features similar to hand-crafted features.

Explainability for the feature-engineering approach to translationese classification. To date, explainability in translationese research has mainly focused on quantifying handcrafted feature importance. Techniques include inspecting SVM feature weights (Avner et al., 2016; Pylypenko et al., 2021), correlation (Rubino et al., 2016), information gain (Ilisei et al., 2010), chi-square (Ilisei et al., 2010), decision trees or random forests (Rubino et al., 2016; Ilisei et al., 2010), ablating features and observing the change in accuracy (Baroni and Bernardini, 2005; Ilisei et al., 2010), training separate classifiers on each individual feature (or feature set) and comparing accuracies (Volansky et al., 2015; Avner et al., 2016). For n-grams, the difference in frequencies between the original and translationese classes (Koppel and Ordan, 2011; van Halteren, 2008), and the contribution to the symmetrized Kullback-Leibler Divergence between the classes (Kurokawa et al., 2009) have been used.

Explainability for the neural approach to translationese classification. To date, explainability methods for neural networks have not been widely explored. Pylypenko et al. (2021) quantify to which extent handcrafted features can explain the variance in the predictions of neural models, such as BERT, LSTMs, and a simplified Transformer, by training per-feature linear regression models to output the predicted probabilities of the neural models and computing the R^2 measure. They find that most of

the top features are either POS-perplexity-based, or bag-of-POS features. However, their method treats the neural network as a black-box, whereas we use a method that accesses the internals of the model.

Integrated Gradients (IG). In our work we use the Integrated Gradients method (Sundararajan et al., 2017) for explainability. This method provides attribution scores for the input with respect to a certain class. IG calculates the integral of gradients of the model F with respect to the input x (token embedding), along the path from a baseline x' (in our case, PAD token embedding) to the input x:

$$\texttt{IntegratedGrads}_i(x) ::= (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha \quad (1)$$

The strength of the Integrated Gradients method is that it satisfies two fundamental axioms (Sensitivity and Implementation Invariance), while many other popular attribution methods, like Gradients (Simonyan et al., 2014), DeepLift (Shrikumar et al., 2017) and LRP (Bach et al., 2015) violate one or both of them. IG also satisfies the completeness axiom, that is, IG is comprehensive in accounting for attributions and does not just to pick the top label (Sundararajan et al., 2017).

3 Experimental Settings

3.1 Data

For our experiments, we use the monolingual German dataset in the Multilingual Parallel Direct Europarl (MPDE) (Amponsah-Kaakyire et al., 2021) corpus. The set contains 42k paragraphs with half of the texts German originals and the other half translations into German from Spanish (see statistics in Appendix A.1). We perform paragraph-level classification with an average length of 80 tokens per training sample.

We additionally use an in-domain Europarl-based heldout corpus of around 30k paragraphs for training language models and n-gram quartile distributions on it. This corpus consists of original German texts only.

3.2 Base Setup

We compare the traditional SVM-based feature engineering approach, which has demonstrated high performance in previous translationese research, to the BERT model known to be very successful for various NLP tasks, including classification. As base setup, we reproduce the models from Pylypenko et al. (2021) for the two architectures and a new baseline:

1. a linear **SVM** on 108-dimensional **handcrafted feature** vectors (with surface, lexical, unigram bag-of-PoS, language modelling and n-gram frequency distribution features[1]). [**handcr.-features+SVM**]
2. a **linear classifier** (BERT classification head, simple linear FFN, except for difference in input dimension) trained on the 108-dimensional **handcrafted feature** vectors. [**handcr.-features+LinearClassifier**]
3. off-the-shelf Google's **pretrained BERT**-base model (12 layers, 768 hidden dimensions, 12 attention heads) which we **fine-tune** on the MPDE corpus for translationese classification. [**pretrained-BERT-ft**]
4. a BERT-base model with the same settings trained **from scratch** on MPDE for translationese classification. [**fromScratch-BERT**]

For 1, we estimate n-gram language models with SRILM (Stolcke, 2002) and do POS-tagging with SpaCy.[2] For 3, we use multilingual BERT (Devlin et al., 2019) (BERT-base-multilingual-uncased), and fine-tune with the *simpletransformers*[3] library. We use a batch size of 32, learning rate of $4 \cdot 10^{-5}$, and the Adam optimiser with epsilon $1 \cdot 10^{-8}$.

To ensure fair and comprehensive treatment, we carefully explore many experiments and variations below: we exchange input features between BERT and SVM architectures by (i) feeding BERT-learned features into SVMs (Section 3.3), handcrafted features into the BERT classification head, and $(ii\text{-}a)$ letting the full BERT architecture learn handcrafted feature vectors used by SVMs and $(ii\text{-}b)$ feeding handcrafted feature vectors as embeddings into the BERT model (Section 3.4).

3.3 SVM Classifier with BERT Features

We train an SVM with linear kernel on the features learnt by the pretrained BERT model fine-tuned on

[1] See (Pylypenko et al., 2021) for the detailed list of features.

[2] https://spacy.io/

[3] github.com/ThilinaRajapakse/simpletransformers

the translationese classification task. We use the output of the BERT pooler, which selects the last layer $[CLS]$ token vector, with linear projection and $tanh$ activation as our feature vector. We use:

1. BERT's 768-dim pooled vector output, [**pretrained-BERT-ft+SVM**]
2. a 108-dim PCA projection of this vector. [**pretrained-BERT-ft+PCA$_{108}$+SVM**]

The PCA projection allows us to match the handcrafted feature vector dimensionality.

3.4 BERT with Handcrafted Features

Apart from feeding hand-crafted feature vectors into a suitably adjusted BERT classification head [**handcr.-features+LinearClassifier**], we carefully design two strategies to force the full BERT architecture use the handcrafted features.

Pretraining on handcrafted feature prediction. First, we train a BERT-base model from scratch on the MPDE dataset to predict the handcrafted features. This regression model [**BERT-reg-full**] takes unmasked text as input and predicts continuous values (the 108 dimension vectors representing handcrafted features originally used in training the SVM). The complete feature vector is predicted at once, and the pretraining is done by minimizing MSE loss between the predicted and the ground truth vector. The weights of this model encode the information of the handcrafted features. With this pretrained model,

1. we freeze the weights, replace the regression head (linear layer predicting 108 features) with a linear classifier (a BERT classification head predicting the original or translationese label) and train the classifier on the MPDE data for translationese classification, [**BERT-r2c-full-frozen**][4]
2. we do not freeze but fine-tune on MPDE for the translationese classification task. [**BERT-r2c-full-ft**]

The comparison between frozen and unfrozen weights is designed to provide us insights on the importance of representation learning in BERT.

We reproduce the same approach as above with a smaller BERT model with only 6 layers instead of 12 [**BERT-reg-half**]. Interestingly, according to the losses when training for predicting the handcrafted features, the smaller BERT-reg-half performs comparably to BERT-reg-full (0.0041136 vs 0.0041148 MSE). We then load the weights of the small 6 layer model into the embedding layer and the first 6 layers of a 12 layer non-pretrained BERT-base model and, similarly as before:

3. we freeze the loaded weights in the first 6 layers and train the remaining 6 layers and classifier on the translationese classification task, [**BERT-r2c-half-frozen**]
4. we do not freeze but fine-tune on the translationese classification task with randomly-initialised weights for the other 6 layers. [**BERT-r2c-half-ft**]

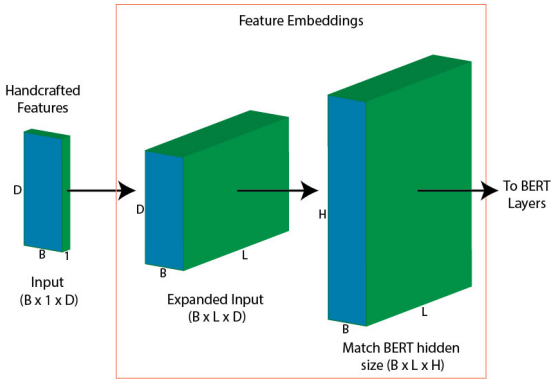

Figure 1: Mapping handcrafted features to embeddings.

Mapping handcrafted features to embeddings. Even though the very low MSE results indicate that both versions of BERT-reg are able to learn handcrafted features well, using them in terms of frozen layers in translationese classification leads to low classification performance (Section 4). This could be attributed to the fact that, not being an end-to-end approach, information losses accumulate: first, even though MSE is low in BERT-reg, we do not have exactly the same features; and second, the features are not used directly for classification, but are encoded again by the network. This motivates us to explore an alternative way of encoding handcrafted features in an end-to-end manner.

We convert the single vector of handcrafted features of dimension D (108 in our experiments) into a sequence of embeddings in BERT's layer format, that is, length of feature embedding sequence L times the dimension of the hidden states H (768), while preserving the information of the single vector (Figure 1). To do this, we consider a batch of

[4] **r2c** – regression-to-classification

Model	Accuracy (%)
handcr.-features+SVM	73.2±0.1
handcr.-features+LinearClassifier	72.0±0.4
pretrained-BERT-ft	92.2±0.2
fromScratch-BERT	89.3±0.3
pretrained-BERT-ft+SVM	92.0±0.0
pretrained-BERT-ft+PCA$_{108}$+SVM	92.0±0.0
BERT-r2c-full-frozen+SVM	74.9±0.7
BERT-r2c-full-frozen+PCA$_{108}$+SVM	70.3±0.1
BERT-r2c-full-frozen	59.6±0.1
BERT-r2c-full-ft	89.3±0.4
BERT-r2c-half-frozen	67.5±0.4
BERT-r2c-half-ft	89.0±0.3
BERT-f2c $L = 1$	57±10
BERT-f2c $L = 80$	72.8±0.2
BERT-f2c $L = 256$	72.7±0.2
pretrained-BERT-f2c $L = 80$	68.0±2.1

Table 1: Translationese classification accuracy for all settings (average and standard deviation over 5 runs). All of the models were trained/fine-tuned for the translationese classification task.

tokens with size B and take in the handcrafted features as a $(B \times D)$-dimensional input to the BERT model and generate feature embeddings by passing the features through 2 linear layers as follows. We first pass the $(B \times 1 \times D)$ input to the first linear layer. The resulting $(B \times L \times D)$-dimensional output is fed as input to the second linear layer which outputs a $(B \times L \times H)$-dimensional output as the feature embeddings.

This reshaped handcrafted feature embedding layer replaces BERT's embedding layer. Weights are randomly initialised and the modified BERT model is trained on the translationese classification task. We experiment with three different values for sequence length L: 1, 80 (average length of our training samples) and 256 (half of maximum input for BERT). All three variants are trained from scratch [**BERT-f2c**[5] **L=1**, **BERT-f2c L=80**, **BERT-f2c L=256**]. For further comparison, we also take BERT-f2c L=80, load the weights of pretrained BERT-base layers into the 12 layers of the modified model and fine-tune on the task **[pretrained BERT-f2c L=80**].

Training and hyperparameter settings for these models are given in Appendix A.2.

4 Translationese Classification

Table 1 summarises results of the different translationese classification settings. For the base models, BERT outperforms the SVM by 16% when trained from scratch and 19% when finetuned.

Feeding pooled output of BERT into the SVM model [**pretrained-BERT-ft+SVM**], accuracy increases by 19% percentage points absolute over using handcrafted features [**handcr.-features+SVM**], even when PCA is used to reduce the BERT vector dimensionality to match the size of the handcrafted feature vector. Feeding handcrafted features directly to the linear BERT classification head [**handcr.-features+LinearClassifier**] reduces accuracy by about 20% points compared to pretrained and fine-tuned BERT [**pretrained-BERT-ft**]. This shows that features learnt by BERT are superior to our set of manual features, as used in previous high performing classical feature engineering-based approaches to translationese classification. When BERT is trained from scratch on the MPDE data [**fromScratch-BERT**], translationese classification accuracy reduces by ~ 3 percentage points, compared to pretrained-BERT-ft. This suggests that pretraining on large data helps to encode additional information that turns out to be helpful in the translationese classification task.

One can assume that BERT pretrained to predict the handcrafted features and subsequently frozen [**BERT-r2c-full-frozen**] has learnt to encode the handcrafted features during pretraining (Section 3.4). Nevertheless, its accuracy, albeit higher than a random guess, is lower by ~ 13 percentage points than the SVM classifier. We perform an additional experiment, in order to check whether the difference in accuracy is due to BERT failing to sufficiently encode the handcrafted features during pretraining, or due to the SVM classifier being superior to the linear classification head of the BERT model. Namely, we train the SVM classifier on the pooled output of BERT-r2c-full-frozen model [**BERT-r2c-full-frozen+SVM**] and on the PCA-reduced dimensionality [**BERT-r2c-full-frozen+PCA$_{108}$+SVM**]. The accuracy is around 75% for both settings which is as high as using SVM on handcrafted feature vectors. We conclude that BERT encodes the handcrafted features sufficiently well, but the linear classifier performs worse than an SVM in these conditions.

Further fine-tuning BERT fully pretrained for handcrafted feature prediction [**BERT-r2c-full-ft**] for translationese classification results in accuracy comparable to BERT that was not pretrained on this task [**fromScratch-BERT**]. This could suggest that our handcrafted feature set is either a sub-

[5] **f2c** – feature-embeddings to classification

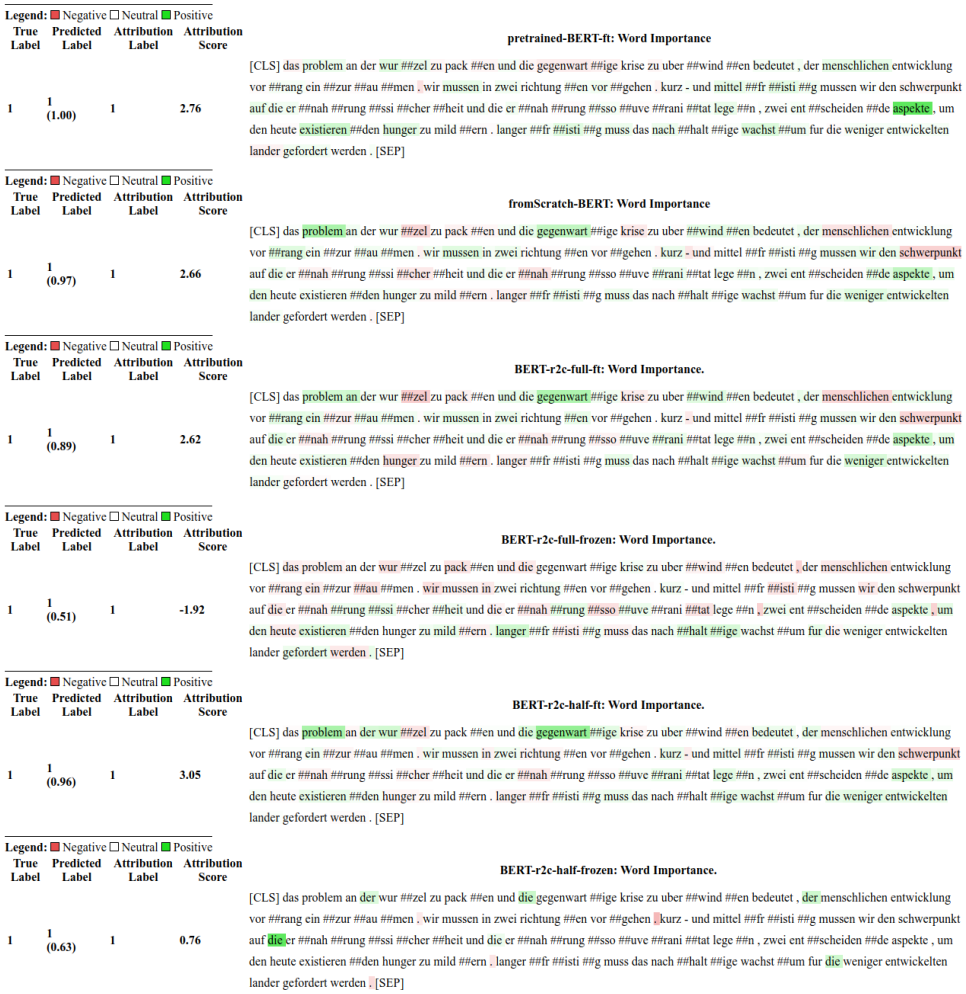

Figure 2: Layer Integrated Gradient saliency maps of input tokens contributing to the ground truth translationese label (here: translation). Comparison of different models.

set of features learned by fromScratch-BERT, or that the handcrafted features are discarded during fine-tuning. The model where only the first 6 layers were pretrained [**BERT-r2c-half-ft**], achieves similar accuracy, likely due to the same reasons. By contrast, freezing the 6 handcrafted feature prediction pretrained layers [**BERT-r2c-half-frozen**] largely reduces the accuracy with respect to BERT-r2c-half-ft, because the model only has access to the 6th layer embeddings that supposedly encode the information about the handcrafted features, and does not have ability to extract its own features. The remaining (higher) 6 layers are responsible for the increment in accuracy with respect to BERT-r2c-full-frozen.

The results of BERT-f2c models show that BERT, when fed the handcrafted features in the form of embeddings, can reach at most the same accuracy as the hancdr.-features+SVM approach, which suggests that the BERT architecture has no advantage over the SVM classifier in utilizing the handcrafted features for classification. This is again evidence that the features, and not the classifier, cause the better performance of the feature and representation learning method.[6]

5 Layer Integrated Gradients Saliency

We compare input attributions of the ground truth classification label amongst **pretrained-BERT-ft**, **fromScratch-BERT** and four different settings of the translationese classification models pretrained on the handcrafted feature prediction task: **BERT-r2c-full-ft**, **BERT-r2c-full-frozen**, **BERT-r2c-half-ft** and **BERT-r2c-half-frozen**. We use Layer Integrated Gradients from the Captum library (Kokhlikyan et al., 2020), which computes the attribution for all the individual neurons in the

[6] As a sanity check, we ran an experiment using a gradient boosting classifier instead of an SVM, with the exact same 108 hand-crafted features and obtain accuracy of 72.3%.

	Translationese				Original			
	BERT-r2c-full-ft		pretrained-BERT-ft		BERT-r2c-full-ft		pretrained-BERT-ft	
Rank	**Token**	**AAS**	**Token**	**AAS**	**Token**	**AAS**	**Token**	**AAS**
1	sagte	0.60	entstand	0.70	##wegen	0.61	situations	0.37
2	gebiet	0.46	virus	0.63	•	0.55	•	0.36
3	##dies	0.44	inti	0.60	eu	0.49	ria	0.34
4	ansicht	0.43	sagte	0.58	daraufhin	0.49	##lk	0.33
5	bezug	0.42	entdeckte	0.57	finde	0.45	##iet	0.32
6	neige	0.40	gras	0.57	##vo	0.45	golden	0.32
7	amt	0.40	nuts	0.56	gerne	0.43	sak	0.30
8	pre	0.40	nicaragua	0.55	##abb	0.42	turm	0.30
9	spanien	0.39	rekord	0.53	##hrte	0.42	##emen	0.27
10	sprechen	0.38	bilbao	0.53	ausbau	0.42	orange	0.27
11	nuts	0.36	verfugte	0.53	!	0.42	hang	0.26
12	barcelona	0.34	bol	0.51	bekommen	0.42	##wald	0.25
13	;	0.33	colombia	0.51	trips	0.41	1732	0.25
14	##bien	0.32	nis	0.51	ez	0.41	dobe	0.24
15	spanischen	0.32	och	0.49	##gemeinde	0.40	##pas	0.23
16	wiederholt	0.31	vorkommen	0.49	vot	0.36	profits	0.22
17	einige	0.30	oecd	0.49	won	0.36	stuttgart	0.22
18	##sprache	0.29	;	0.46	geplant	0.35	soja	0.21
19	weder	0.29	erklarte	0.45	demnach	0.35	r	0.21
20	territorium	0.28	clinton	0.45	ja	0.35	ruth	0.21

Table 2: Top-20 tokens with highest average attribution score (AAS) towards original and translationese classes in the test set. BERT-r2c-full-ft and pretrained-BERT-ft.

embedding layer, and calculate the salience score for each token by averaging the attributions over the embedding dimension.

Comparing Models. Figure 2 displays Integrated Gradients attributions for a translated paragraph across different BERT models. The trends for the original paragraph are similar to those that we observe for the translated paragraph, therefore attributions for the original paragraph are given in Appendix A.3.

Comparing the attributions of classification labels to sample inputs amongst the various settings of BERT, we observe that attributions are similar for **fromScratch-BERT** and the fine-tuned models: **BERT-r2c-full-ft** and **BERT-r2c-half-ft**. This suggests that fine-tuning "dissolves" the pre-learned information about the hand-crafted features in the **r2c** models, no matter how much of the model was pre-trained. By contrast, freezing the weights in **BERT-r2c-full-frozen** and **BERT-r2c-half-frozen** resulted in very different attributions compared to the **fromScratch-BERT**. Since these frozen models only utilize the information they have learnt about the handcrafted features, this shows that this information is not identical to the information that **fromScratch-BERT** learns for the translationese classification task. For **BERT-r2c-half-frozen** the attributions are more peaked than for other models, with only a few tokens receiving large scores, and most tokens having scores close to zero. Notably, **pretrained-BERT-ft** displays a pattern that is overall similar to BERT trained from scratch, but some attributions are reversed, and the peaks are on different tokens. This supports the observation that off-the-shelf BERT pretrained on a large amount of data encodes some useful additional information.

For **BERT-r2c-full-frozen**, a substantial number of tokens with negative attributions have positive attributions in the model trained from scratch and also the fine-tuned models. However some attributions overlap, which suggests that **fromScratch-BERT** may be using something like the handcrafted features. We investigate this further by examining the fine-tuning checkpoints.

Comparing Checkpoints. We aim to study how **fromScratch-BERT** learns information about translationese classification over the epochs, and how this compares to the fine-tuning of **BERT-r2c-full-ft**, when the information about the handcrafted features is gradually modified over the epochs turn into the final feature set used for translationese classification. In Appendix A.3 we provide additional results on examining training checkpoints for **fromScratch-BERT** and **BERT-r2c-full-ft** for an original and a translated paragraph.

Results indicate that for **fromScratch-BERT**

some attributions change into their opposite during training, whereas for **BERT-r2c-full-ft** the pattern appears to be already settled from the early checkpoints onwards, and does not change much over the course of fine-tuning. This supports the hypothesis that the handcrafted features are a subset of features learnt by **fromScratch-BERT**, and thus provide a useful initialization of weights for fine-tuning for translationese classification.

Highest Average Attribution. In order to make the interpretation less local, and to generalize the observations, we compute the top tokens with highest attribution on average across the test set. The results for each class for best-performing models (**pretrained-BERT-ft** and **BERT-r2c-full-ft**) are given in Table 2.

For German translationese data translated from Spanish, some top tokens correspond to the geographical areas, where Spanish is spoken, e.g. "spanien", "barcelona", "spanischen" for BERT-r2c-full-ft; "nicaragua", "colombia", "bilbao" for pretrained-BERT-ft. (Moreover, in this example it appears that off-the-shelf pretrained-BERT-ft, pretrained on the Wikipedia data, better utilizes the non-European toponyms, unlike the BERT-r2c-full-ft that was only trained on the European-focused Europarl data.) Likewise for original German data, some of the top tokens are German geographical names, e.g. "stuttgart" for pretrained-BERT-ft. The subword "##wald" also appears to be a common German toponymic suffix. This suggests that topic is one of the spurious clues that is used by BERT to determine the correct translationese class. This is also supported by the fact that some nouns that likely correspond to certain recurring discussion topics for only one class within our data sample, receive high attribution, e.g. "virus", "soja", "clinton", "orange" etc. The "ez" token, salient for the original class, appears to be a starting subword unit of the *EZB* abbreviation (Europäische Zentralbank).

The "•" token (bullet point) having a high attribution for the class *originals* for both models might suggest a spurious correlation within the dataset, that is apparently utilized by BERT. The ";" token is deemed important for the translationese class by both models, which might also be a spurious correlation. Conversely, this could be an indication that clauses in Spanish are more often joint with the semi-colon, than in German, which was preserved in the translation. This corroborates findings from other works that deep networks exploit spurious statistical cues for better performance (Mudrakarta et al., 2018; Niven and Kao, 2019).

For both models the Präteritum forms "sagte", "erklärte" etc. are also among the top tokens important for recognizing translationese. One possible explanation could be that the Perfekt form ("hat gesagt") is more common in German spoken language, and Präteritum is more common in writing. Therefore the translators, while translating Spanish speeches into German, may have preferred to use the Präteritum form more common for writing.

6 Summary and Conclusions

We address two open questions in classification-based translationese research: (1) are the substantial performance differences between feature- and representation-learning and classical handcrafted feature based approaches is due to (i) the difference in the features, (ii) the classifiers, or (iii) both, and (2) what do feature- and representation-learning based approaches actually learn?

We address (1) by exchanging features from both models, examining a broad variety of settings, to ensure that this is done in a fair an unbiased way. We show that SVMs perform as good as BERT when fed with features learnt by BERT. Likewise, the BERT classification head and the full BERT architecture perform at the level of traditional SVM-based classification with handcrafted features, when fed with handcrafted features only. This shows that it is the feature and representation learning and not the classifiers that are responsible for the translationese classification performance difference.

To address question (2), we examine BERT's input attributions using Integrated Gradients Saliency for various settings and observe that attributions are indeed similar for the model trained from scratch (fromScratch-BERT) on just the text data and the fine-tuned models that were pretrained on handcrafted feature prediction (BERT-r2c-full-ft and BERT-r2c-half-ft). This suggests that pretraining on the handcrafted features does not make a visible difference in attributions, and, together with the accuracy result that also does not change, suggests that no extra information is learnt during pretraining on handcrafted features. Based on these findings, and the fact that some attributions appear to overlap for BERT pretrained on handcrafted features and where the pretrained layers were subse-

quently frozen (BERT-r2c-full-frozen), and BERT trained from scratch (fromScratch-BERT), it is consistent to assume that handcrafted features are a (possibly partial) subset of the features automatically learnt by BERT.

Finally, analysis of top activated tokens suggests that at least part of BERT's strong translationese classification accuracy is based on topic differences between the classes as well as on some spurious correlations, rather than "proper" translationese phenomena. We are currently working on quantifying the 'Clever Hans' behaviour using named entity masking and cleaning/normalizing the data.

Acknowledgements

We would like to thank the reviewers for their insightful comments and feedback. This research is funded by the German Research Foundation (Deutsche Forschungsgemeinschaft) under grant SFB 1102: Information Density and Linguistic Encoding.

References

Kwabena Amponsah-Kaakyire, Daria Pylypenko, Cristina España-Bonet, and Josef van Genabith. 2021. Do not rely on relay translations: Multilingual parallel direct Europarl. In *Proceedings for the First Workshop on Modelling Translation: Translatology in the Digital Age*, pages 1–7, online. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2020. Translation artifacts in cross-lingual transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7674–7684, Online. Association for Computational Linguistics.

Ehud Alexander Avner, Noam Ordan, and Shuly Wintner. 2016. Identifying translationese at the word and sub-word level. *Digit. Scholarsh. Humanit.*, 31:30–54.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10:e0130140.

Mona Baker, Gill Francis, and Elena Tognini-Bonelli. 1993. Corpus linguistics and translation studies: Implications and applications. In *Text and Technology: In Honour of John Sinclair*, page 233–, Netherlands. John Benjamins Publishing Company.

Marco Baroni and Silvia Bernardini. 2005. A New Approach to the Study of Translationese: Machine-learning the Difference between Original and Translated Text. *Literary and Linguistic Computing*, 21(3):259–274.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Markus Freitag, Isaac Caswell, and Scott Roy. 2019. APE at scale and its implications on MT evaluation biases. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 34–44, Florence, Italy. Association for Computational Linguistics.

Martin Gellerstam. 1986. Translationese in Swedish novels translated from English. *Translation studies in Scandinavia*, 1:88–95.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2020. Statistical power and translationese in machine translation evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 72–81, Online. Association for Computational Linguistics.

Iustina Ilisei, Diana Inkpen, Gloria Corpas Pastor, and Ruslan Mitkov. 2010. Identification of translationese: A machine learning approach. In *Computational Linguistics and Intelligent Text Processing*, pages 503–511, Berlin, Heidelberg. Springer Berlin Heidelberg.

Anders Kaas, Viktor Torp Thomsen, and Barbara Plank. 2020. Team DiSaster at SemEval-2020 task 11: Combining BERT and hand-crafted features for identifying propaganda techniques in news. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1817–1822, Barcelona (online). International Committee for Computational Linguistics.

Amirmohammad Kazameini, Samin Fatehi, Yash Mehta, Sauleh Eetemadi, and Erik Cambria. 2020. Personality trait detection using bagged SVM over BERT word embedding ensembles. *CoRR*, abs/2010.01309.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for pytorch.

Moshe Koppel and Noam Ordan. 2011. Translationese and its dialects. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1318–1326, Portland, Oregon, USA. Association for Computational Linguistics.

David Kurokawa, Cyril Goutte, and Pierre Isabelle. 2009. Automatic detection of translated text and its impact on machine translation. In *Proceedings of Machine Translation Summit XII: Papers*, Ottawa, Canada.

Wah Meng Lim and Harish Tayyar Madabushi. 2020. UoB at SemEval-2020 task 12: Boosting BERT with corpus level information. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2216–2221, Barcelona (online). International Committee for Computational Linguistics.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906, Melbourne, Australia. Association for Computational Linguistics.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Anushka Prakash and Harish Tayyar Madabushi. 2020. Incorporating count-based features into pre-trained models for improved stance detection. In *Proceedings of the 3rd NLP4IF Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 22–32, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

Daria Pylypenko, Kwabena Amponsah-Kaakyire, Koel Dutta Chowdhury, Josef van Genabith, and Cristina España-Bonet. 2021. Comparing feature-engineering and feature-learning approaches for multilingual translationese classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8596–8611, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ella Rabinovich and Shuly Wintner. 2015. Unsupervised identification of translationese. *Transactions of the Association for Computational Linguistics*, 3:419–432.

Biswarup Ray and Avishek Garain. 2020. Factuality classification using bert embeddings and support vector machines. In *IberLEF@SEPLN*.

Parker Riley, Isaac Caswell, Markus Freitag, and David Grangier. 2020. Translationese as a language in "multilingual" NMT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7737–7746, Online. Association for Computational Linguistics.

Raphael Rubino, Ekaterina Lapshinova-Koltunski, and Josef van Genabith. 2016. Information density and quality estimation features as translationese indicators for human translation classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 960–970, San Diego, California. Association for Computational Linguistics.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3145–3153. JMLR.org.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*.

Jasdeep Singh, Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. XLDA: Cross-Lingual Data Augmentation for Natural Language Inference and Question Answering. *ArXiv*, abs/1905.11471.

Ilia Sominsky and Shuly Wintner. 2019. Automatic detection of translation direction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1131–1140, Varna, Bulgaria. INCOMA Ltd.

Andreas Stolcke. 2002. SRILM – An extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, pages 901–904.

Sara Stymne. 2017. The effect of translationese on tuning for statistical machine translation. In *The 21st Nordic Conference on Computational Linguistics*, pages 241–246.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org.

Elke Teich. 2003. *Cross-Linguistic Variation in System und Text. A Methodology for the Investigation of Translations and Comparable Texts*. Mouton de Gruyter, Berlin.

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.

Gideon Toury. 1980. *In Search of a Theory of Translation*. The Porter Institute for Poetics and Semiotics, Tel Aviv University, Tel Aviv.

Hans van Halteren. 2008. Source language markers in EUROPARL translations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 937–944, Manchester, UK. Coling 2008 Organizing Committee.

Vered Volansky, Noam Ordan, and Shuly Wintner. 2015. On the features of translationese. *Digital Scholarship in the Humanities*, 30(1):98–118.

Cheng Zhang and Hayato Yamana. 2020. WUY at SemEval-2020 task 7: Combining BERT and naive Bayes-SVM for humor assessment in edited news headlines. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1071–1076, Barcelona (online). International Committee for Computational Linguistics.

Mike Zhang and Antonio Toral. 2019. The effect of translationese in machine translation test sets. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 73–81, Florence, Italy. Association for Computational Linguistics.

A Appendix

A.1 Extra Information on the MPDE Dataset

We use version 2.0.0 of the MPDE dataset licensed under CC-BY 4.0. Specifically we use the *mono_de_es* train/dev/test splits of the German-Spanish language pair. Table 3 contains summary statistics of the data.

Split	Number of Examples
Train set	29580
Validation set	6366
Test	6344

Table 3: Dataset statistics.

A.2 Extra Information on BERT Models

With the exception of pretrained-BERT-ft, we use the *transformers* library.[7] Training is done across 4 NVIDIA GeForce GTX TITAN X GPUs with a batch size of 8 per GPU. We use a learning rate of $3 \cdot 10^{-5}$ and train or fine-tune for 5 epochs. Table 4 shows the number of parameters of the different BERT variants. Parameter counts include the embedding and respective prediction (classifier or regression) layers.

Model	Num. Params (M)
fromScratch-BERT	177.85
BERT-reg-full	177.94
BERT-reg-half	135.41
BERT-r2c-*	177.85
BERT-f2c $L = 1$	177.46
BERT-f2c $L = 80$	177.52
BERT-f2c $L = 256$	177.66
pretrained-BERT-f2c $L = 80$	177.52

Table 4: Number of parameters of the various BERT models.

[7] https://huggingface.co/transformers/model_doc/bert.html

A.3 Additional Layer Integrated Gradients saliency maps

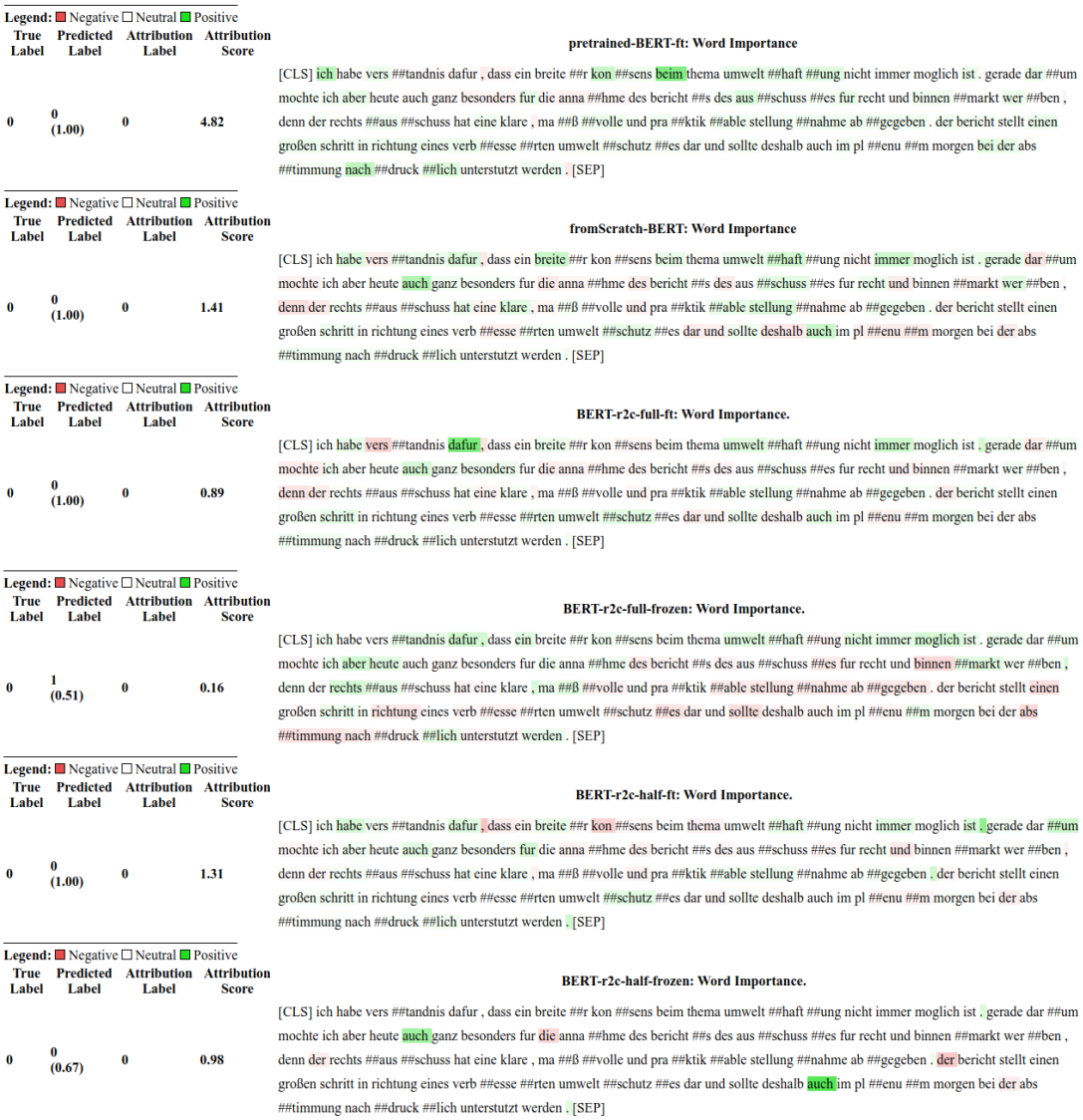

Figure 3: Layer Integrated Gradient saliency maps of input tokens contributing to the ground truth translationese label (here: original). Comparison of different models.

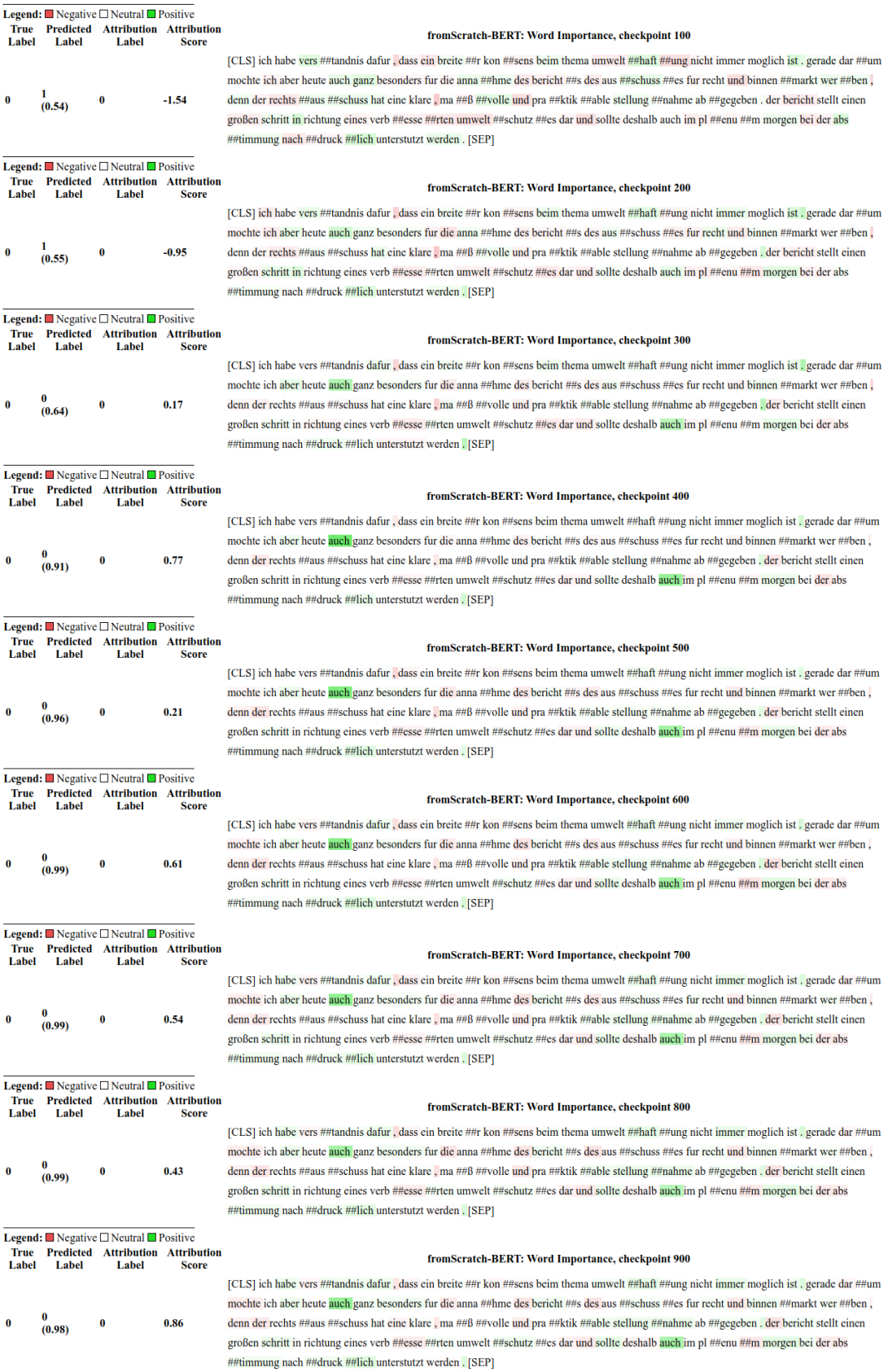

Figure 4: Layer Integrated Gradient saliency maps of input tokens contributing to the ground truth translationese label (here: original). BERT trained from scratch for translationese classification. Changes in attribution over the training checkpoints.

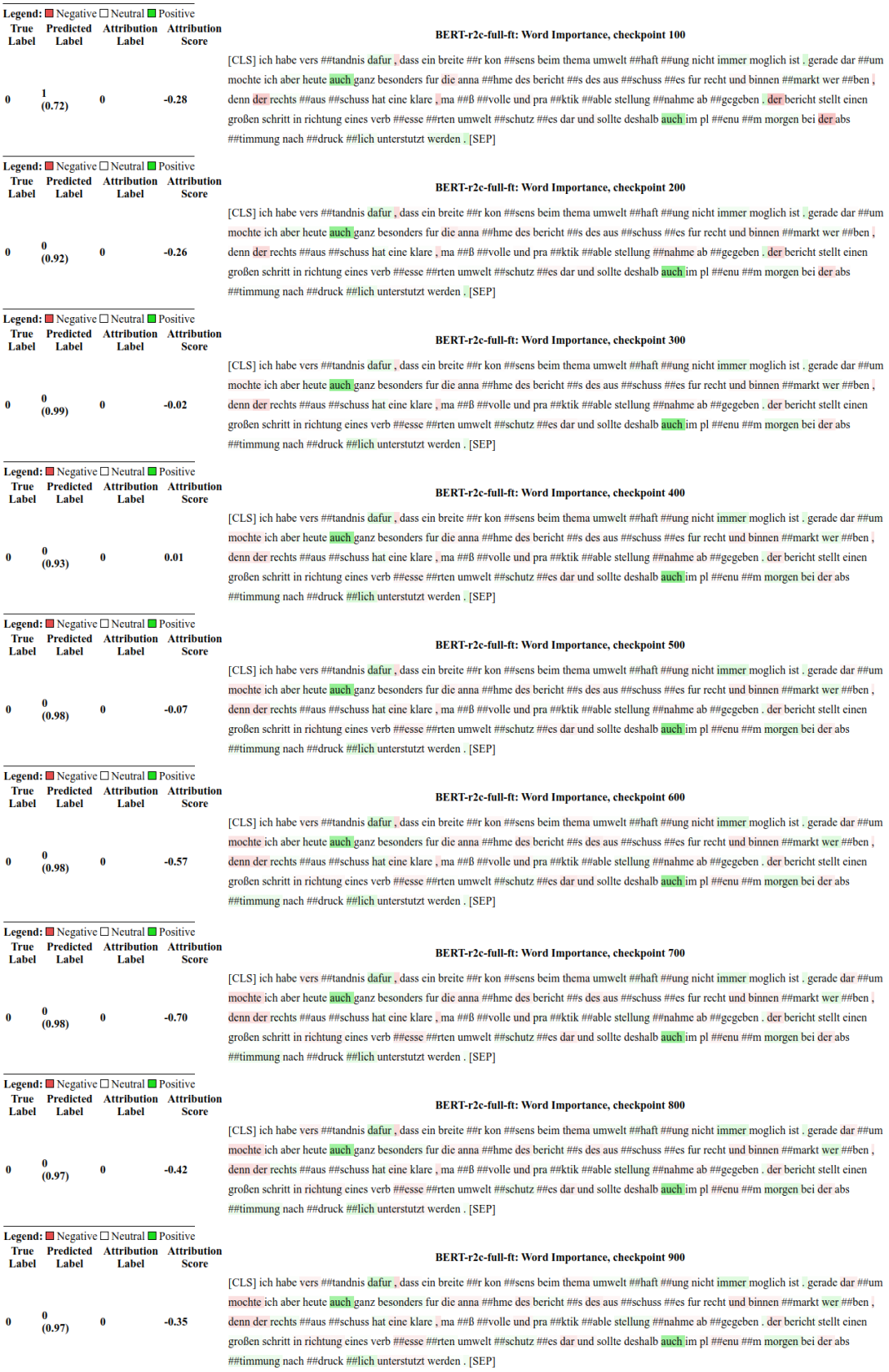

Figure 5: Layer Integrated Gradient saliency maps of input tokens contributing to the ground truth translationese label (here: original). BERT pretrained for handcrafted feature prediction, and fine-tuned for translationese classification. Changes in attribution over the training checkpoints.

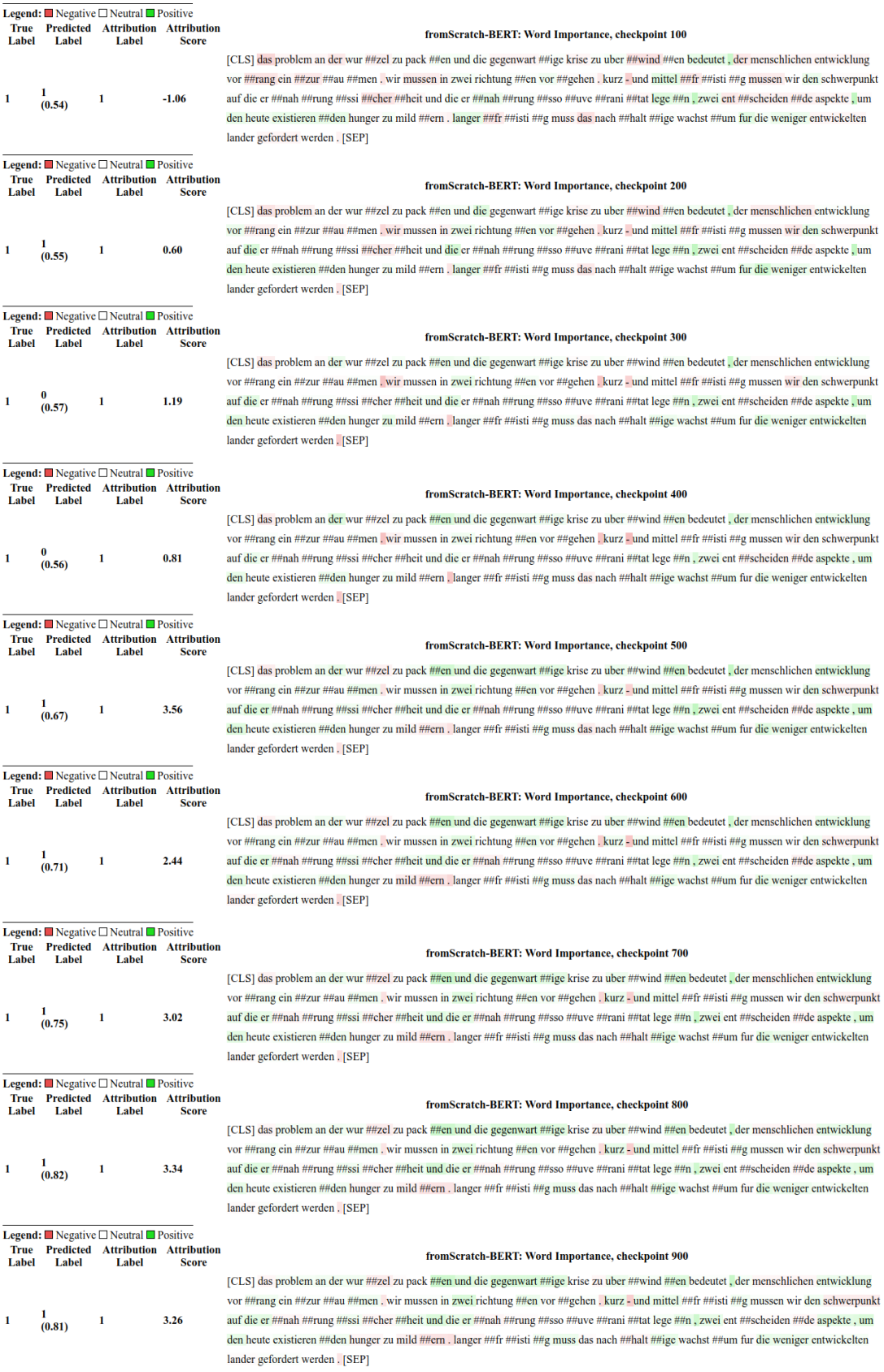

Figure 6: Layer Integrated Gradient saliency maps of input tokens contributing to the ground truth translationese label (here: translation). BERT trained from scratch for translationese classification. Changes in attribution over the training checkpoints.

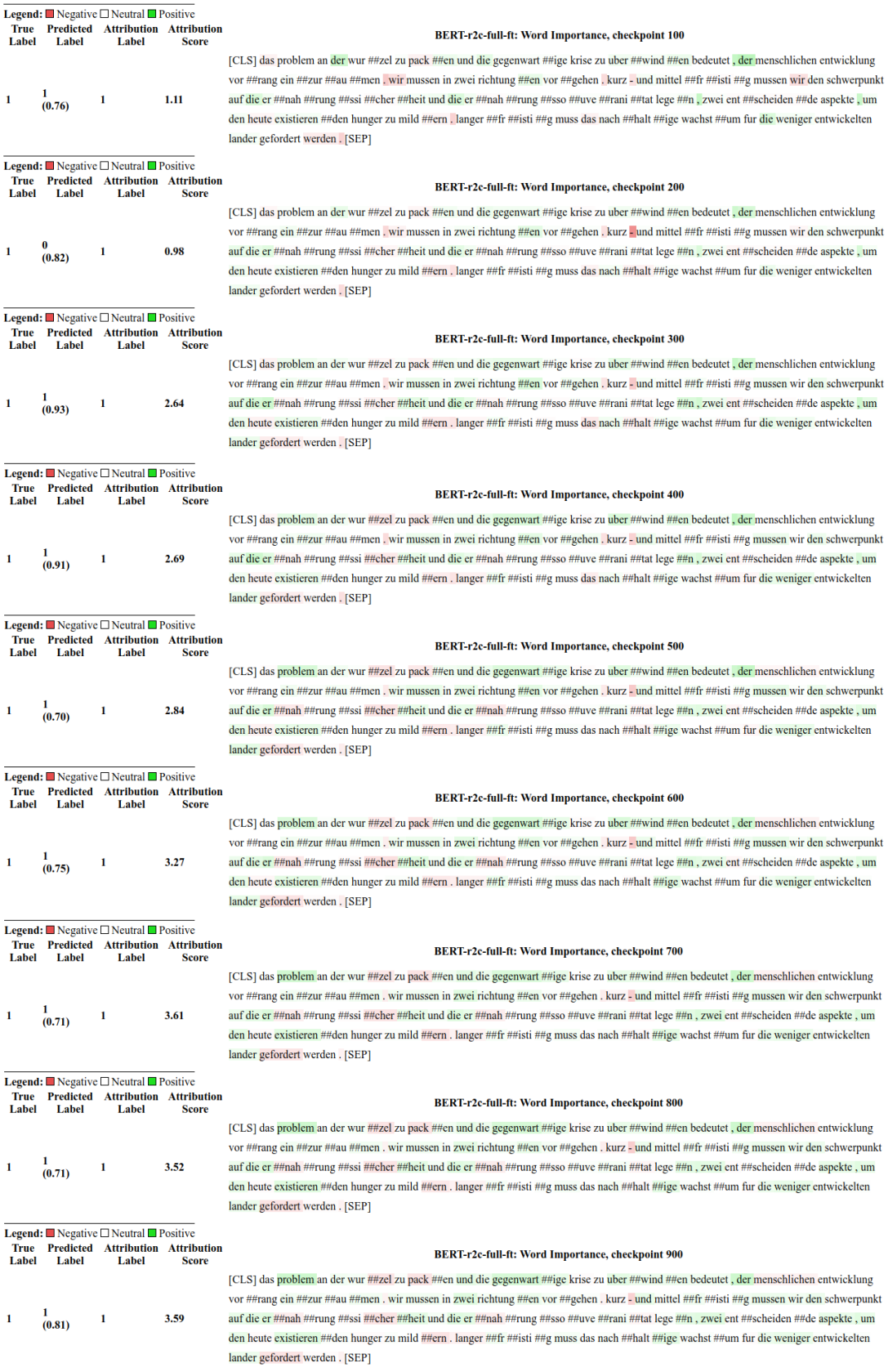

Figure 7: Layer Integrated Gradient saliency maps of input tokens contributing to the ground truth translationese label (here: translation). BERT pretrained for handcrafted feature prediction, and fine-tuned for translationese classification. Changes in attribution over the training checkpoints.