Investigating and Explaining Feature and Representation Learning in Translationese Classification

Anonymous ACL submission

Abstract

Recent work has shown that neural feature- and representation-learning approaches, and specifically the BERT model, demonstrates superior performance over traditional manual feature engineering and an SVM classifier for the task of translationese classification for various source and target languages. However, to date it is unclear whether the performance differences are due to better representations, better classifiers or both. Moreover, it remains unclear whether the features learnt by BERT overlap with commonly used manual features. To answer these, we exchange features between BERT-based and SVM classifiers, and show that, an SVM fed with BERT representations performs at the level of the best BERT classifiers, and BERT learning and using hand-crafted features performs at the level of traditional classifiers using hand-crafted features. Our experiments indicate that our hand-crafted feature set does not provide any additional information that BERT has not learnt already, and is likely to be a subset of features automatically learnt by BERT. Finally, we apply Integrated Gradients to examine token importance for the BERT model, and find that part of its top performance results are due to just 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, and even human experts may not be able to reliably distinguish between original texts and professional translations (Tirkkonen-Condit, 2002), corpus-based studies (Baker et al., 1993) and, in particular, machine-learning classifier based studies (Rabinovich and Wintner, 2015; Volansky et al., 2015; Rubino et al., 2016; Pylypenko et al., 2021) clearly reveal the differences. 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. 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) shows that feature- and representation-learning deep neural network-based approaches (in particular BERT-based, but also other neural approaches) to translationese 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, or the combination of both, and (ii) what the neural feature and representation learning approaches actually learn. The contributions of our paper are as follows:

1. we address (i) by feeding BERT-based learned features to SVMs and by letting BERT architectures learn handcrafted features, as well as feeding the handcrafted features into BERT as embeddings. Our experiments show that SVMs using BERT-learned features perform on a par with our best BERT-

translationese classifiers. Moreover, BERT using handcrafted features only performs at most as good as the SVM classifier.

2. we present the first steps to address (ii) using attribution-based explainable AI approaches (XAI) on our best performing full feature- and representation-learning BERT model and on BERT models that are pre-trained to predict handcrafted features and then fine-tuned for translationese classification. We present evidence that 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 (source texts in Spanish translated to English e.g. may have a higher likelihood of talking about Spanish places). Moreover, some top features suggest that there might be certain spurious correlations within our dataset.

To the best of our knowledge this is the first paper that shows that feature- and representation-learning rather than the classifier is responsible for the substantial performance gap between deep neural networks and machine learning approaches using handcrafted features. It is also the first paper that uses XAI methods to (begin to) explain what neural methods learn in translationese classification.

Finally, translationese research is not just an "academic" exercise in basic research into aspects of how translation works, but an important research topic in machine translation evaluation (Stymne, 2017; Toral et al., 2018; Freitag et al., 2019; Edunov et al., 2020; Graham et al., 2020) and in further improving machine translation (Riley et al., 2020).

2 Related Work

(Kaas et al., 2020; Prakash and Tayyar Madabushi, 2020; Lim and Madabushi, 2020) combine BERT-based and handcrafted features in an ensemble manner in order to improve over BERT's accuracy, often by concatenating the pooled output of BERT with a handcrafted feature vector (sometimes additionally encoded by another network) and feeding them into another classifier. They show that even though BERT representations are powerful, carefully picked handcrafted features may still provide additional information that aids the task in hand. We also aim to verify if there is an overlap between the features learnt by BERT and our handcrafted feature set. In some cases the classifier used for ensembling BERT and handcrafted features is a Support Vector Machine (Kazameini et al., 2020; Ray and Garain, 2020).

Explainability methods for neural networks have not been widely explored for translationese classification. Since many previous works have used the traditional feature-engineering method, they often quantify handcrafted feature importance. Techniques used for that include looking at 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, then 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.

As for looking into the neural network performance, Pylypenko et al. (2021) quantify whether hand-crafted features can explain the variance in the predictions of neural models, such as BERT, LSTM, and 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, this method treats the neural network as a black-box, whereas we use a method that accesses the internals of the model.

In our work we use the Integrated Gradients method (Sundararajan et al., 2017) the method provides attribution scores for the input with respect to a certain class. It involves calculating the integral of gradients with respect to the input along the path from a certain baseline (in our case, PAD tokens) to the input.

3 Experimental Settings

3.1 Data

For our experiments, we use the monolingual German dataset in the Multilingual Parallel Direct Eu-

roparl (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. The average length is 80 tokens per training sample. Since there exists a problem with pivot translations in Europarl (Bogaert, 2011), the DE-ES dataset contains only data from before 2004, when the pivot system was introduced. We additionally use a heldout corpus of around 30k paragraphs for estimating language models and n-gram quartile distributions. This corpus consists of originally produced texts only. For the heldout corpus, we sample texts from Europarl proceedings from 2004 onwards, since original data is not affected by the pivot translation problem.

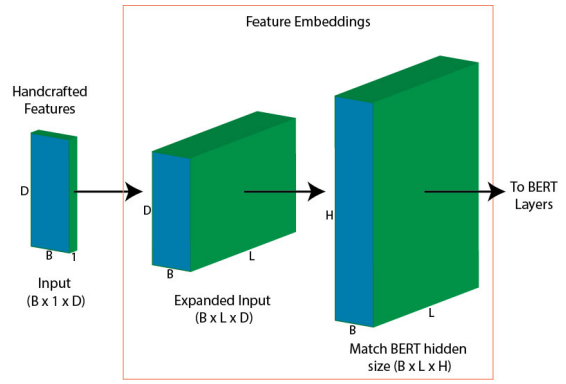

Figure 1: Mapping handcrafted features to embeddings (Section 3.4).

3.2 Base Setup

A fair comparison between learned and handcrafted features and two classifiers is non-trivial. We feed learned features into SVMs, and (i) let BERT learn handcrafted feature vectors used by SVMs and (ii) feed handcrafted feature vectors as embeddings into the BERT model. We compare this with full feature and representation learning settings with BERT, and SVMs with handcrafted features. To test this for translationese classification we reproduce the models from Pylypenko et al. (2021):

1. a linear SVM with 108 handcrafted features (with surface, lexical, unigram bag-of-PoS, language modelling and n-gram frequency distribution features), [**handcr.-features+SVM**]

2. a pretrained BERT-base model (12 layers, 768 hidden dimensions, 12 attention heads) fine-tuned on translationese classification. [**pretrained-BERT-ft**]

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

3.3 SVM Classifier with BERT Features

We train a SVM with linear kernel on the features learnt by the pretrained BERT model fine-tuned on 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-output+SVM**]

2. a 108-dim PCA projection of the vector. [**pretrained-BERT-ft-output+PCA(108 dim) +SVM**]

The later allows us to match the handcrafted feature vector dimensionality.

3.4 BERT with Handcrafted Features

Making neural networks use handcrafted features in our comparison is non-trivial. We design two strategies.

Pretraining on handcrafted feature prediction. First, we train a BERT-base model from scratch, using it to predict the 108 dimension vectors representing handcrafted features originally used in training the SVM [**BERT-reg-full**]. The weights of this model encode the information of the handcrafted features. With this pretrained model,

1. we freeze the weights and train a classifier on top for translationese classification, [**BERT-r2c-full-frozen**]

2. we do not freeze but fine-tune on the translationese classification task. [**BERT-r2c-full-ft**]

We explore the same with a smaller BERT model with only 6 layers instead of 12 [**BERT-reg-half**]. 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:

[1] github.com/ThilinaRajapakse/simpletransformers

[2] https://spacy.io/

Model	Test accuracy (%)
handcr.-features + SVM	73.2±0.1
pretrained-BERT-ft-output + PCA(108 dim) + SVM	92.0±0.0
pretrained-BERT-ft-output + SVM	92.0±0.0
BERT-r2c-full-frozen-output + PCA(108 dim) + SVM	70.3±0.1
BERT-r2c-full-frozen-output + SVM	74.9±0.7
pretrained-BERT-ft	92.2±0.2
fromScratch-BERT	89.3±0.3
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 $L = 1$	57.1±10.1
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.

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 it on the translationese classification task with randomly-initialised weights for the other 6. [**BERT-r2c-half-ft**]

Interestingly, according to the losses when training for predicting the handcrafted features, BERT-reg-half performs comparably to the BERT-reg-full (0.0041136 vs 0.0041148).

We also train a BERT-base model with the same settings from scratch on the translationese classification task as a baseline for all BERT models. [**fromScratch-BERT**].

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. This motivates us to explore an alternative way of encoding handcrafted features: we convert the single vector of handcrafted features of dimension D (108 in our experiments) into a sequence of embeddings in BERT's layers 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. (Fig. 1)

To do this, we consider a batch of 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 unsqueeze the $B \times D$ input to $B \times 1 \times D$ dimensions and reshape it as $B \times D \times 1$. This is passed to the first linear layer. The resulting $B \times D \times L$-dimensional output is reshaped as $B \times L \times D$ and fed as input to the second linear layer which outputs a $B \times L \times H$-dimensional output as the feature embeddings.

This hand-crafted feature embedding replaces BERT's embedding layer and serves as input to the first BERT layer. The resulting BERT model is trained on the translationese classification task. We experiment with three different values for L: 1, 80 (average length of our training samples) and 256 (maximum input for BERT). [**BERT-f2c L=1**, **BERT-f2c L=80**, **BERT-f2c L=256**]

4 Translationese Classification

Table 1 summarises results of the different translationese classification settings. As for feeding pooled output of BERT into the SVM model, we can observe that the accuracy is a lot higher comparing to feeding handcrafted features, even when the BERT vector dimensionality is reduced to match the amount of handcrafted features. This emphasizes the fact that the features learnt by BERT are superior to our current set of manual features.

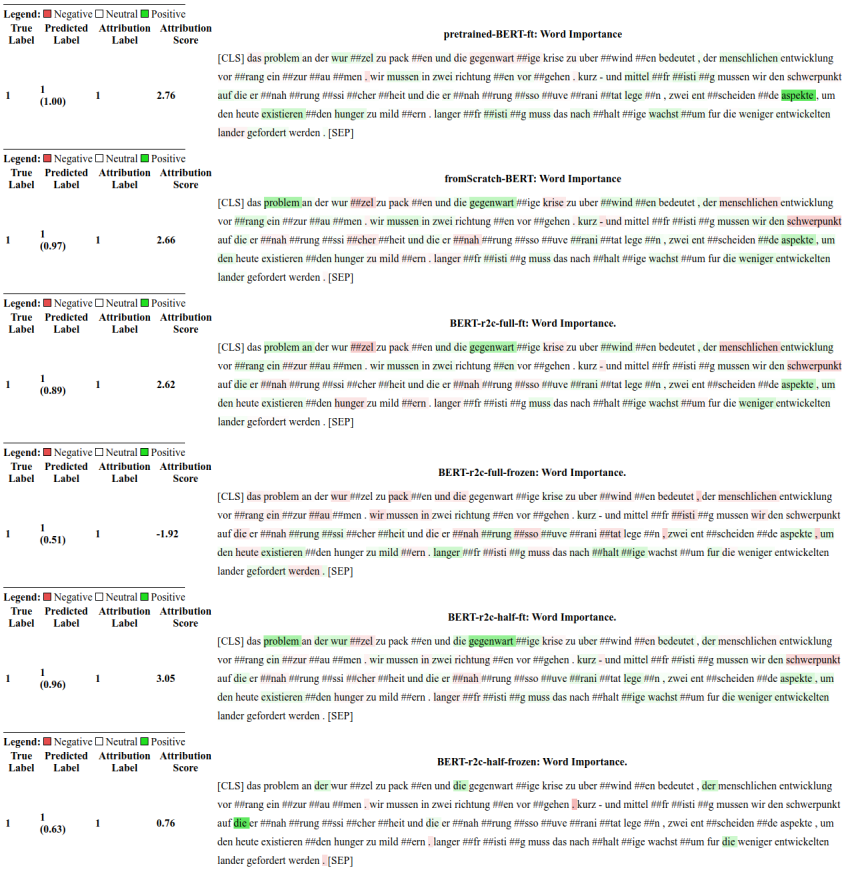

Figure 2: Layer Integrated Gradient saliency maps of input tokens contributing to the ground truth translationese label (here: translation). Comparison of different models.

When BERT is trained from scratch, translationese classification accuracy reduces by 2.93 percentage points, compared to the 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.

As for BERT pretrained to predict the handcrafted features and frozen (BERT-r2c-full-frozen), it is assumed that BERT has learnt to encode the handcrafted features during pretraining. Nevertheless, its accuracy, albeit higher than a random guess, is lower by $\sim$ 13 percentage points comparing to 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. The accuracy is around 75% (for both original and PCA-reduced dimensionality) which is as high as using SVM on handcrafted feature vectors. We conclude that BERT encodes the handcrafted features sufficiently enough, but the linear classifier performs worse than an SVM in these conditions, possibly due to non-exhaustive hyperparameter search.

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 subset 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, because the model only has access to the 6th layer embeddings that supposedly encode only the information about the handcrafted features, and does not have ability to extract its own features from the input text, due to its inability to tune the embeddings.

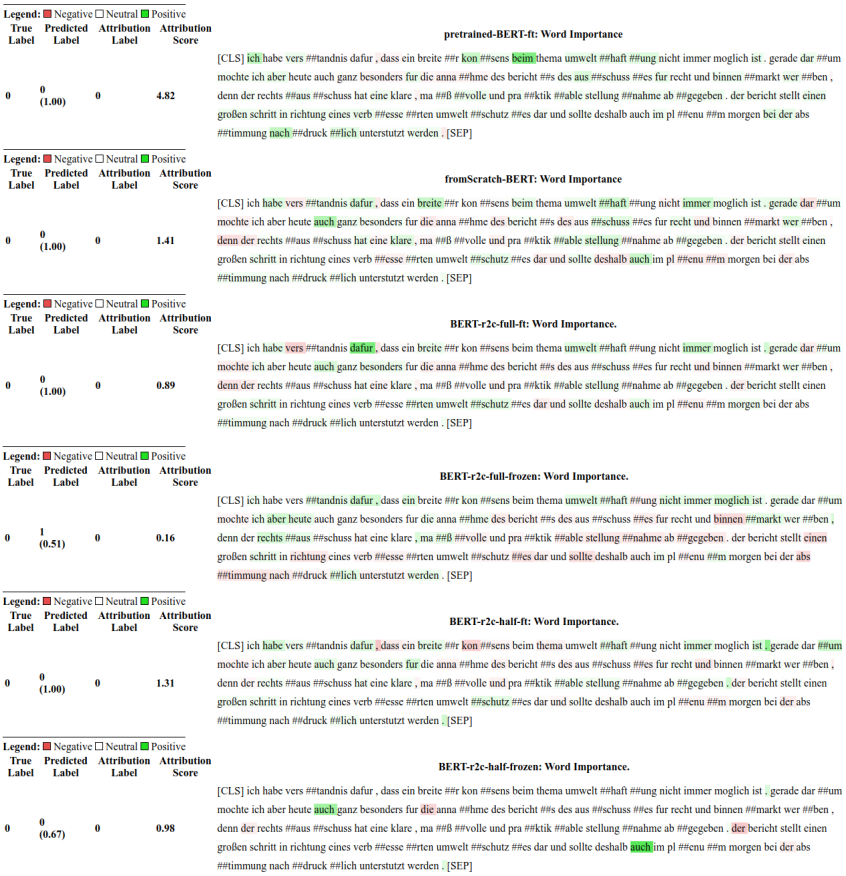

Figure 3: Layer Integrated Gradient saliency maps of input tokens contributing to the ground truth translationese label (here: original). Comparison of different models.

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.

5 Layer Integrated Gradients Saliency

We compare input attributions of the ground truth classification label amongst pretrained-BERT-ft, fromScratch-BERT and the different settings of the translationese classification models pretrained on the hand-crafted feature prediction task. We use Layer Integrated Gradients from the Captum library (Kokhlikyan et al., 2020), and calculate the salience score for each token by averaging the attributions over the embedding dimension.

5.1 Comparing Models

Figure 2 displays Integrated Gradient attributions for a **translated** paragraph across different BERT models. Figure 3 shows attributions for an **original** paragraph.

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. By contrast, freezing the weights in BERT-r2c-full-frozen and BERT-r2c-half-frozen resulted in very different attributions from the fromScratch-BERT. 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 the BERT trained from scratch, but some attributions are reversed, and the peaks are on different tokens.

For BERT-r2c-full-frozen, it can be seen that 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 BERT might be using some of the handcrafted features.

BERT-r2c-full-ft			pretrained-BERT-ft		
Rank	Token	Avg attribution score	Rank	Token	Avg attribution score
1	sagte	0.60	1	entstand	0.70
2	gebiet	0.46	2	virus	0.63
3	##dies	0.44	3	inti	0.60
4	ansicht	0.43	4	sagte	0.58
5	bezug	0.42	5	entdeckte	0.57
6	neige	0.40	6	gras	0.57
7	amt	0.40	7	nuts	0.56
8	pre	0.40	8	nicaragua	0.55
9	spanien	0.39	9	rekord	0.53
10	sprechen	0.38	10	bilbao	0.53
11	nuts	0.36	11	verfugte	0.53
12	barcelona	0.34	12	bol	0.51
13	;	0.33	13	colombia	0.51
14	##bien	0.32	14	nis	0.51
15	spanischen	0.32	15	och	0.49
16	wiederholt	0.31	16	vorkommen	0.49
17	einige	0.30	17	oecd	0.49
18	##sprache	0.29	18	;	0.46
19	weder	0.29	19	erklarte	0.45
20	territorium	0.28	20	clinton	0.45

Table 2: Top-20 tokens with highest average attribution score towards the **translationese** class in the test set. BERT-r2c-full-ft and pretrained-BERT.

5.2 Comparing Checkpoints

In Appendix A 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 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 may support the hypothesis that the handcrafted features are a subset of features learnt by BERT, and thus provide a favorable initialization of weights for fine-tuning for translationese classification.

5.3 Highest Average Attribution

In order to make the interpretation less local, 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 and BERT-r2c-full-ft) are given in Tables 2 and 3.

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. Likewise for original German data, some of the top tokens are German geographical names, e.g. "stuttgart" for pretrained BERT. 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 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.

For both models the Präteritum forms "sagte", "erklärte" etc. are also among the top tokens impor-

BERT-r2c-full-ft			pretrained-BERT-ft		
Rank	Token	Avg attribution score	Rank	Token	Avg attribution score
1	##wegen	0.61	1	situations	0.37
2	•	0.55	2	•	0.36
3	eu	0.49	3	ria	0.34
4	daraufhin	0.49	4	##lk	0.33
5	finde	0.45	5	##iet	0.32
6	##vo	0.45	6	golden	0.32
7	gerne	0.43	7	sak	0.30
8	##abb	0.42	8	turm	0.30
9	##hrte	0.42	9	##emen	0.27
10	ausbau	0.42	10	orange	0.27
11	!	0.42	11	hang	0.26
12	bekommen	0.42	12	##wald	0.25
13	trips	0.41	13	1732	0.25
14	ez	0.41	14	dobe	0.24
15	##gemeinde	0.40	15	##pas	0.23
16	vot	0.36	16	profits	0.22
17	won	0.36	17	stuttgart	0.22
18	geplant	0.35	18	soja	0.21
19	demnach	0.35	19	r	0.21
20	ja	0.35	20	ruth	0.21

Table 3: Top-20 tokens with highest average attribution score towards the **original** class in the test set. BERT-r2c-full-ft and pretrained-BERT.

tant 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, could have preferred to use the Präteritum form more common for writing.

6 Conclusion

This paper addresses 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 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. We confirm that SVMs perform as good as BERT when fed with features learnt by BERT. Likewise, BERT performs at the level of traditional SVM-based classification with handcrafted features SVMs, when fed with handcrafted features only. Our findings reveal that while pretraining on huge amount of data improves the classification accuracy, pretraining on handcrafted features does not guarantee an improvement on classification accuracy with respect to training from scratch.

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 and the fine-tuned models that were pretrained on handcrafted feature prediction.

Finally, analysis of top activated tokens in the test set suggests that at least part of BERT's strong translationese classification accuracy is based on topical differences between the classes (rather than "proper" translationese phenomena), the topical differences between the classes, and spurious correlations. The next step would be to control these factors, for instance by using named entity masking and cleaning/normalizing the corpus, in order to investigate whether BERT would still outperform the traditional approach under such conditions.

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.

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

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.

Caroline Bogaert. 2011. Is absolute multilingualism maintainable? The language policy of the European Parliament and the threat of English as a lingua franca. Master's thesis, UGent. Faculteit Letteren en Wijsbegeerte.

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.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. 2020. On the evaluation of machine translation systems trained with back-translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2836–2846, Online. 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 P. Isabelle. 2009. Automatic detection of translated text and its impact on machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, Maryland.

Wah Meng Lim and Harish Tayyar Madabushi. 2020. Uob at semeval-2020 task 12: Boosting BERT with corpus level information. *CoRR*, abs/2008.08547.

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.

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.

Sonja Tirkkonen-Condit. 2002. Translationese – a myth or an empirical fact?: A study into the linguistic identifiability of translated language. *Target. International Journal of Translation Studies*, 14(2):207–220.

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.

A Appendix

A.1 Extra Information on 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 4 contains summary statistics of the data.

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

Table 4: Dataset statistics

A.2 Extra Information on BERT models

With the exception of pretrained-BERT-ft, we use the *transformers* library.[3] 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 5 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 5: Number of parameters of the various BERT models

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

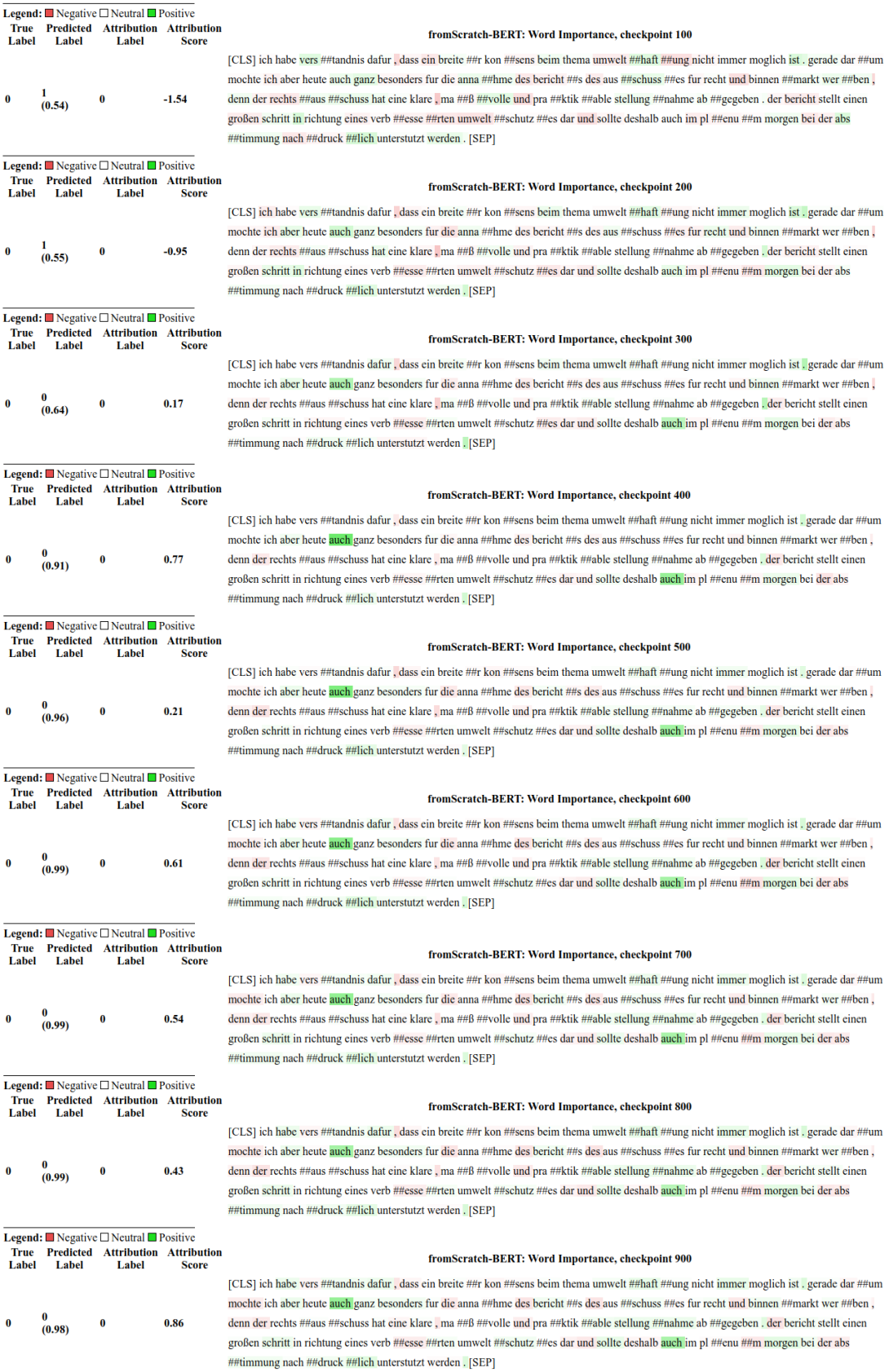

Figure 4: Layer Integrated Gradient saliency maps of input tokens contributing to the ground truth translationese label (here: original). BERT trained from scratch for translationese classification. Changes in attribution over the training checkpoints.

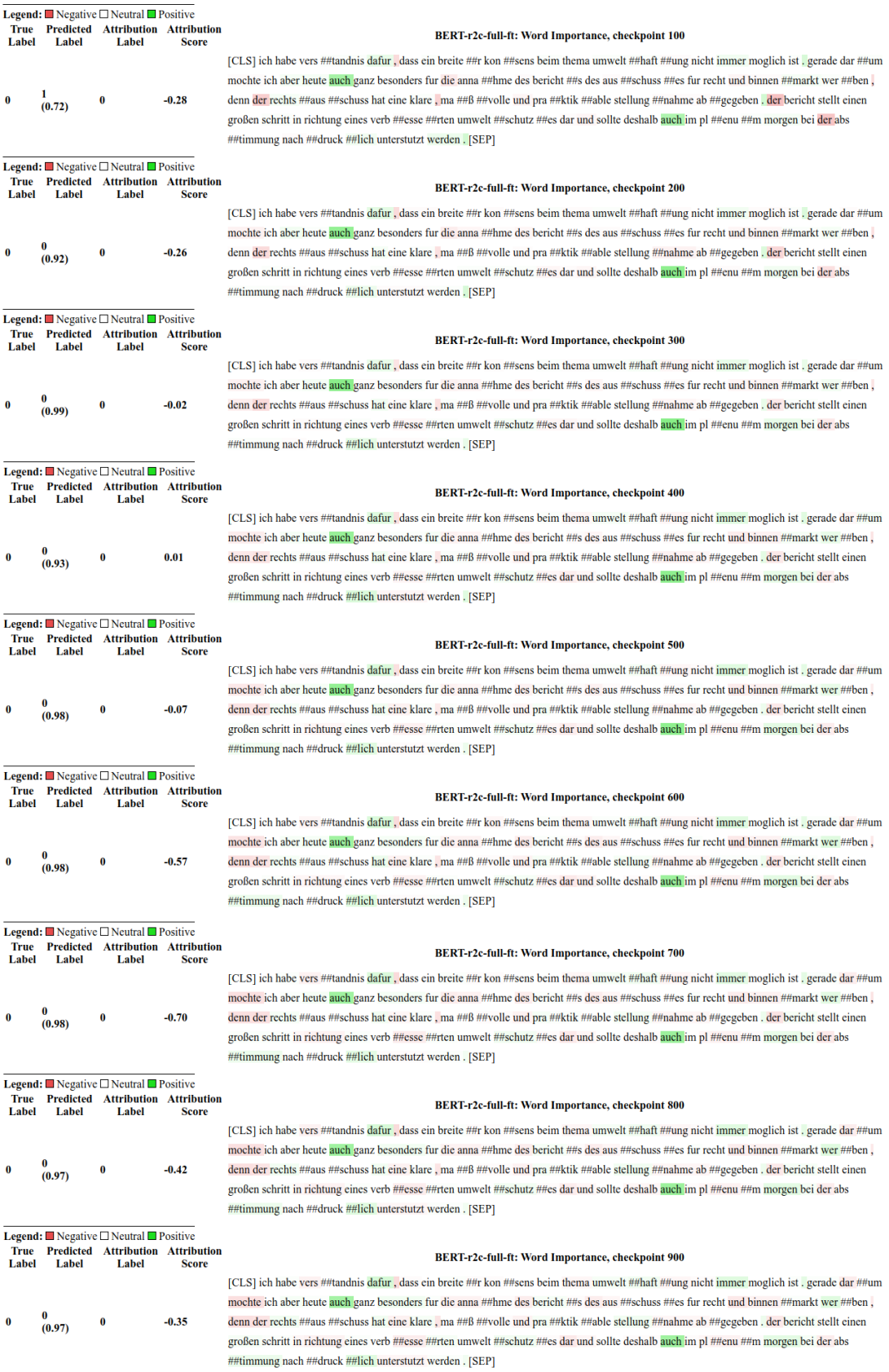

Figure 5: Layer Integrated Gradient saliency maps of input tokens contributing to the ground truth translationese label (here: original). BERT pretrained for handcrafted feature prediction, and fine-tuned for translationese classification. Changes in attribution over the training checkpoints.

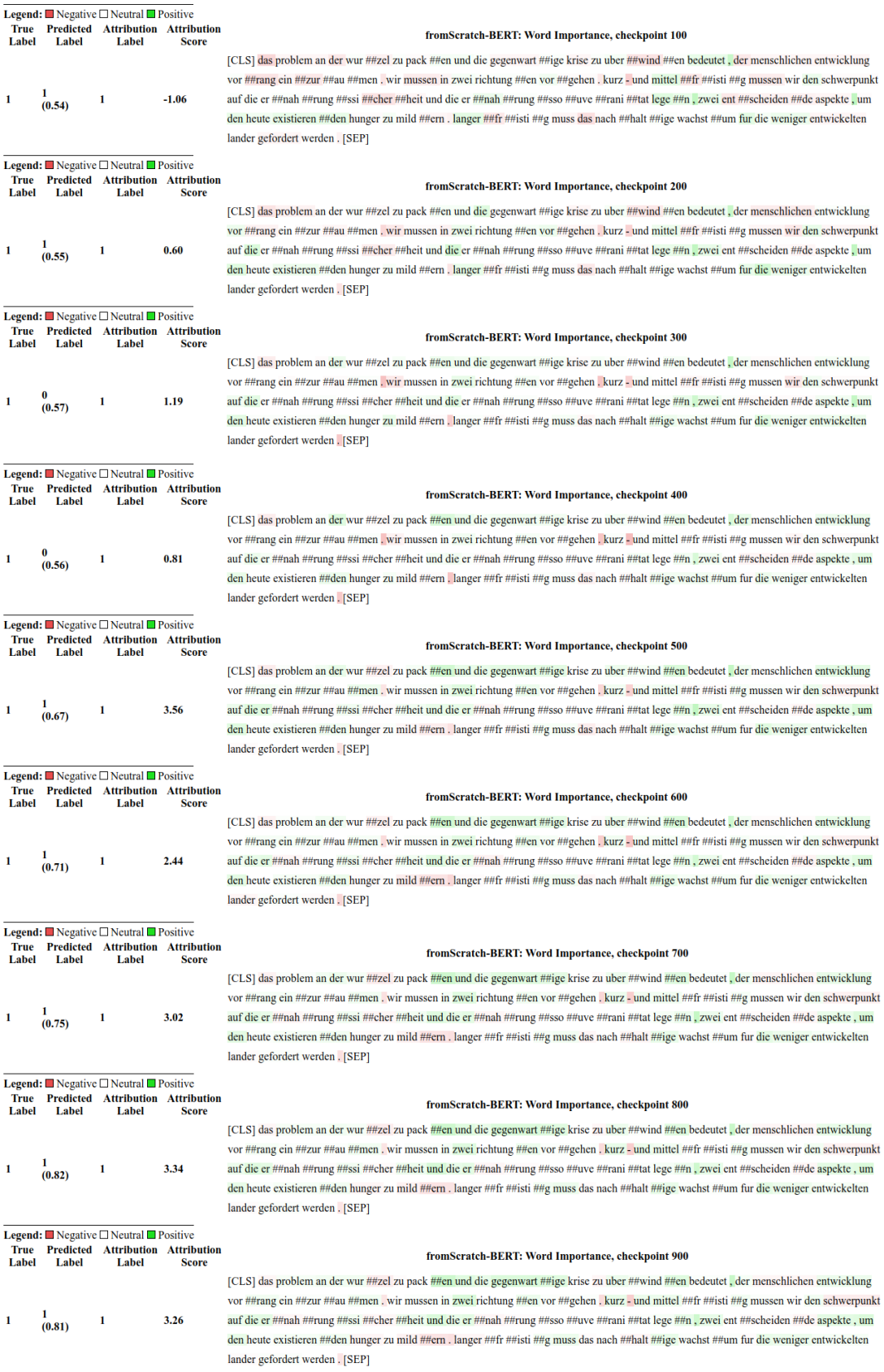

Figure 6: Layer Integrated Gradient saliency maps of input tokens contributing to the ground truth translationese label (here: translation). BERT trained from scratch for translationese classification. Changes in attribution over the training checkpoints.

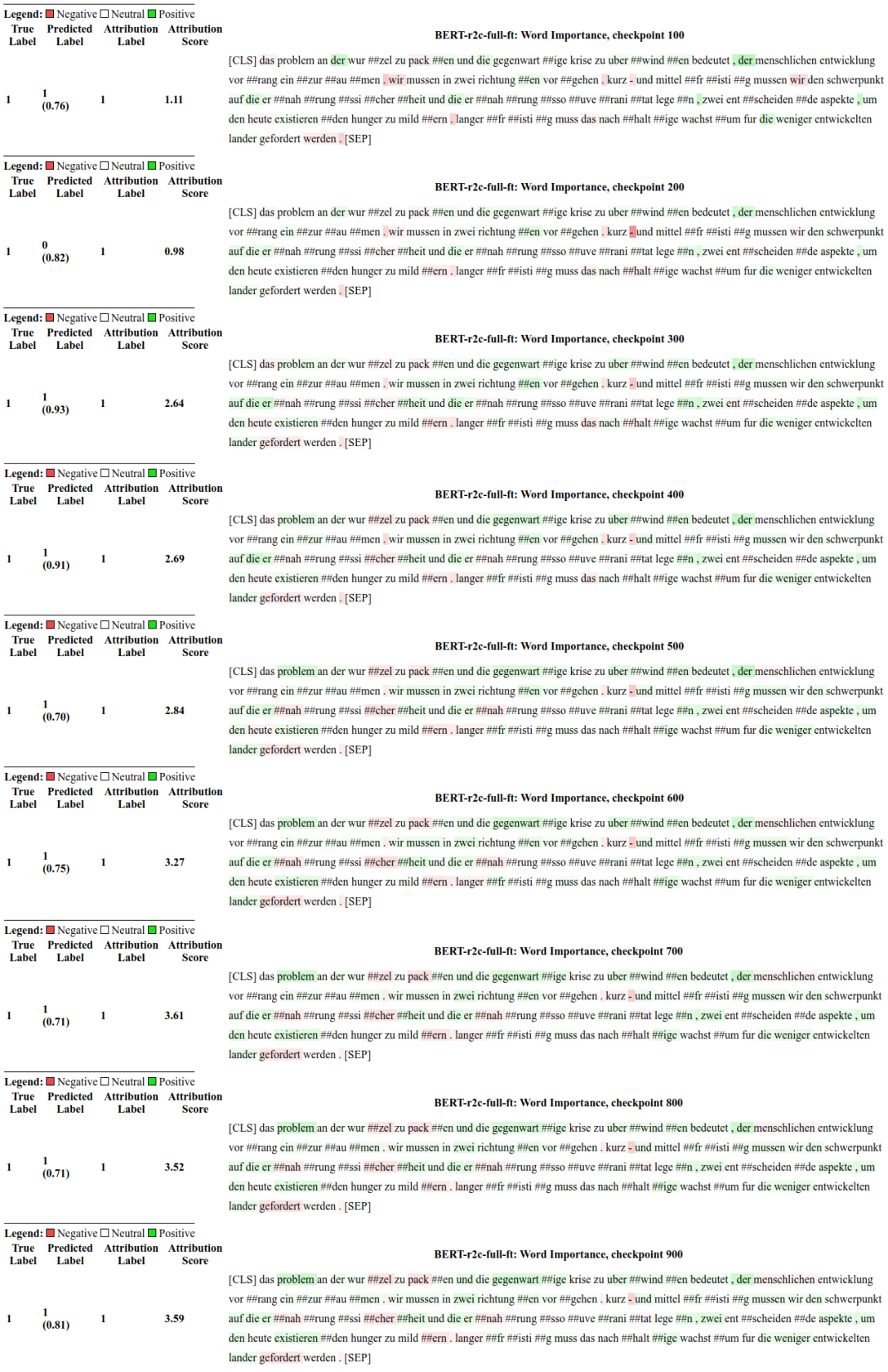

Figure 7: Layer Integrated Gradient saliency maps of input tokens contributing to the ground truth translationese label (here: translation). BERT pretrained for handcrafted feature prediction, and fine-tuned for translationese classification. Changes in attribution over the training checkpoints.