Automatic Discrimination of Human and Neural Machine Translation

van der Werff, Tobias; van Noord, Rik; Toral, Antonio

Published in:
Proceedings of the 23rd Annual Conference of the European Association for Machine Translation

Automatic Discrimination of Human and Neural Machine Translation: A Study with Multiple Pre-Trained Models and Longer Context

Tobias van der Werff
Bernoulli Institute
University of Groningen
t.n.van.der.werff@student.rug.nl

Rik van Noord
CLCG
University of Groningen
rikvannoord@gmail.com

Antonio Toral
CLCG
University of Groningen
a.toral.ruiz@rug.nl

Abstract

We address the task of automatically distinguishing between human-translated (HT) and machine translated (MT) texts. Following recent work, we fine-tune pre-trained language models (LMs) to perform this task. Our work differs in that we use state-of-the-art pre-trained LMs, as well as the test sets of the WMT news shared tasks as training data, to ensure the sentences were not seen during training of the MT system itself. Moreover, we analyse performance for a number of different experimental setups, such as adding *translationese* data, going beyond the sentence-level and normalizing punctuation. We show that (i) choosing a state-of-the-art LM can make quite a difference: our best baseline system (DEBERTA) outperforms both BERT and ROBERTA by over 3% accuracy, (ii) adding *translationese* data is only beneficial if there is not much data available, (iii) considerable improvements can be obtained by classifying at the document-level and (iv) normalizing punctuation and thus avoiding (some) shortcuts has no impact on model performance.

1 Introduction

Generally speaking, translations are either performed manually by a human, or performed automatically by a machine translation (MT) system. There exist many use cases in Natural Language Processing in which working with a human-translated text is not a problem, as they are usually of high quality, but in which we would like to filter out automatically translated texts. For example, consider training an MT system on a parallel corpus crawled from the Internet: we would preferably only keep the high-quality human-translated sentences.

In this paper, we will address this task of discriminating between human-translated (HT) and machine-translated texts automatically. Studies that have analysed MT outputs and HTs comparatively have found evidence of systematic differences between the two (Ahrenberg, 2017; Vanmassenhove et al., 2019; Toral, 2019). These outcomes provide indications that an automatic classifier should in principle be able to discriminate between these two classes, at least to some extent.

There is previous related work in this direction (Arase and Zhou, 2013; Aharoni et al., 2014; Li et al., 2015), but they used Statistical Machine Translation (SMT) systems to get the translations, while the introduction of Neural Machine Translation (NMT) has considerably improved general translation quality and has led to more natural translations (Toral and Sánchez-Cartagena, 2017). Arguably, the discrimination between MT and HT is therefore more difficult with NMT systems than it was with previous paradigms to MT.

We follow two recent publications that have attempted to distinguish NMT outputs from HTs (Bhardwaj et al., 2020; Fu and Nederhof, 2021) and work with MT outputs generated by state-of-the-art online NMT systems. Additionally, we also build a classifier by fine-tuning a pre-trained language model (LM), given the fact that this approach obtains state-of-the-art performance in many text-based classification tasks.

The main differences with previous work are:

- We experiment with state-of-the-art LMs, instead of only using BERT- and ROBERTA-based LMs;
- We empirically check the performance impact of adding *translationese* training data;
- We go beyond sentence-level by training and testing our best system on the document-level;
- We analyse the impact of punctuation shortcuts by normalizing the input texts;
- We use the test sets of WMT news shared task as our data sets, to ensure reproducibility and that the MT system did not see the translations during its training.

The rest of the paper is organised as follows. Section 2 outlines previous work on the topic. Section 3 details our methodology, focusing on the data sets, classifiers and evaluation metrics. Subsequently, Section 4 presents our experiments and their results. These are complemented by a discussion and further analyses, in Section 5. Finally, Section 6 presents our conclusions and suggestions for future work. All our data, code and results is publicly available at `https://github.com/tobiasvanderwerff/HT-vs-MT`

2 Related Work

Analyses Previous work has dealt with finding systematic and qualitative differences between HT and MT. Ahrenberg (2017) compared manually an NMT system and a HT for one text in the translation direction English-to-Swedish. They found that the translation by NMT was closer to the source and exhibited a more restricted repertoire of translation procedures than the HT. Related, an automatic analysis by Vanmassenhove et al. (2019) found that translations by NMT systems exhibit less lexical diversity than HTs. A contemporary automatic analysis corroborated the finding about less lexical diversity and concluded also that MT led to translation that had lower lexical density, were more normalised and had more interference from the source language (Toral, 2019).

SMT vs HT classification Given these findings, it is no surprise that automatic classification to discriminate between MT and HT has indeed been attempted in the past. Most of this work targets SMT since it predates the introduction of NMT and uses a variety of approaches. For example, Arase and Zhou (2013) relied on fluency features, while Aharoni et al. (2014) used part-of-speech tags and function words, and Li et al. (2015) parse trees, density and out-of-vocabulary words. Their methods reach quite high accuracies, though indeed rely on SMT systems, which are of considerable lower quality than the current NMT ones.

NMT vs HT classification To the best of our knowledge only two publications have tackled this classification with the state-of-the-art paradigm, NMT (Bhardwaj et al., 2020; Fu and Nederhof, 2021). We now outline these two publications and place our work with respect to them.

Bhardwaj et al. (2020) work on automatically determining if a French sentence is HT or MT, with the source sentences in English. They test a variety of pre-trained language models, either multilingual –XLM-R (Conneau et al., 2020) and mBERT (Devlin et al., 2019a)– or monolingual for French: CamemBERT (Martin et al., 2020) and FlauBERT (Le et al., 2020). Moreover, they test their trained models across different domains and MT systems used during training. They find that pre-trained LMs can perform this task quite well, with accuracies of over 75% for both in-domain and cross-domain evaluation. Our work follows theirs quite closely, though there are a few important differences. First, we use publicly available WMT data, while they use a large private data set, which unfortunately limits reproducibility. Second, we analyze the impact of punctuation-type "shortcuts", while it is unclear to what extent this gets done in Bhardwaj et al. (2020).[1] Third, we also test our model on the document-level, instead of just the sentence-level.

Fu and Nederhof (2021) work on the WMT18 news commentary data set for translating Czech, German and Russian into English. By fine-tuning BERT they obtain an accuracy of 78% on all languages. However, they use training sets from WMT18, making it highly likely that Google Translate (which they use to get the translations) has seen these sentences during training.[2] This means that the MT outputs they get are likely of higher quality than it would be the case in a

[1] They do apply 12 conservative regular expressions, but, as there is no code available, it is unclear what these are and what impact this had on their results.

[2] This likely does not apply to Bhardwaj et al. (2020), as they use a private data set.

real-world scenario, and thus closer to HT, which would make the task unrealistically harder for the classifiers. On the other hand, an accuracy of 78% is quite high on this challenging task, so perhaps this is not the case. This accuracy might even be suspiciously high: it could be that the model overfit on the Google Translations, or that the data contains artifacts that the model uses as a shortcut.

Original vs MT Finally, there are three related works that attempt to discriminate between MT and original texts written in a given language, rather than human translations as is our focus. Nguyen-Son et al. (2019a) tackles this by matching similar words within paragraphs and subsequently estimating paragraph-level coherence. Nguyen-Son et al. (2019b) approaches this task by round-trip translating original and machine-translated texts and subsequently using the similarities between the original texts and their round-trip translated versions. Nguyen-Son et al. (2021) extends the former work improving the detection of MT even if a different system is used.

3 Method

3.1 Data

We will experiment with the test sets from the WMT news shared tasks.[3] We choose this data set mainly for these four reasons:

(i) it is publicly available so it guarantees reproducibility;

(ii) it has the translation direction annotated, hence we can inspect the impact of having original text or human-translated text (i.e. *translationese*) in the source side;

(iii) the data sets are also available at the document-level, meaning we can train and evaluate systems that go beyond sentence-level;

(iv) these sets are commonly used as test sets, so it is unlikely that they are used as training data in online MT systems, which we use in our experiments.

We will use the German-English data sets, and will focus on the translation direction German-to-English. This language pair has been present the longest at WMT's news shared task, from 2008 till the present day. Hence, it is the language pair with the most test data available. We use 2008 to 2017 as training, 2018 as dev and 2019 as test. Full statistics are shown in Table 1.

[3]For example, `https://www.statmt.org/wmt20/translation-task.html`

Data set	# SNT$_O$	# SNT$_T$	# DOC$_O$	# DOC$_T$
WMT08	361	0	15	0
WMT09	432	448	17	21
WMT10	500	505	15	22
WMT11	601	598	16	18
WMT12	611	604	14	18
WMT13	500	500	7	9
WMT14	1,500	1,503	96	68
WMT15	736	1,433	33	48
WMT16	1,499	1,500	87	68
WMT17	1,502	1,502	66	64
WMT18 (dev)	1,498	—	69	—
WMT19 (test)	2,000	—	145	—
WMT08-17	8,242	8,593	366	336
WMT14-17	5,237	5,938	282	248

Table 1: Statistics of the data sets. # SNT stands for number of sentences, # DOC for number of documents, *O* for number of sentences or documents in which the source side is original, while *T* stands for *translationese*. WMT08-17 and WMT14-17 indicate the sizes of the two training sets used.

Translationese For each of these sets, roughly half of the data was originally written in our source language (German) and human-translated to our target language (English), while the other half was originally written in our target language (English) and translated to our source language (German) by a human translator. We thus make a distinction between text that originates from text written in the source language (German), and text that originates from a previous translation (i.e. English to German). We will refer to the latter as *translationese*.

Half of the data can thus be considered a different category: the source sentences are actually not original, but a translation, which means that the machine-translated output will actually be an automatic translation of a human translation, instead of an automatic translation of original text. In that part of the data, the texts in the HT category are not human translations of original text, but the original texts themselves. Since this data might exhibit different characteristics, given that the translation direction is the inverse, we only use the sentences and documents that were originally written in German for our dev and test sets (indicated with *O* in Table 1). Moreover, we empirically evaluate in Section 4 whether removing the extra *translationese* data from the training set is actually beneficial for the classifier.

MT Since we are interested in contrasting HT vs state-of-the-art NMT, we automatically translate the sentences using a general-purpose and widely used online MT system, DeepL.[4] We translate from German to British English,[5] specifically. We use this MT system for the majority of our experiments, though we do experiment with cross-system classification by testing on data that was translated with other MT systems, such as Google Translate, using their paid API.[6] We manually went through a subset of the translations by both DeepL and Google Translate and indeed found them to be of high quality.

To be clear, in our experiments, the machine translations actually double the size of the train, dev and test sets as indicated in Table 1. For each German source sentence, the data set now contains a human translation (HT, taken from WMT) and a machine translated variant (MT, from DeepL or Google), which are labelled as such. As an example, if we train on both the *original* and *translationese* sentence-level data of WMT08-17, we actually train on $8,242 \cdot 2 + 8,593 \cdot 2 = 33,670$ instances. Note that this also prevents a bias in topic or domain towards either HT or MT.

Ceiling To get a sense of what the upper ceiling performance of this task will be, we check the number of cases where the machine translation is the exact same as the human translation. For DeepL, this happened for 3.0% of the WMT08-17 training set sentences, 3.1% of the dev set and 3.9% of the test set. For Google, the percentages are 2.4%, 2.0% and 3.5%, respectively.[7] Of course, in practice, it is likely impossible to get anywhere near this ceiling, as the MT system also sometimes offers arguably better translations (see Section 5 for examples).

[4] `https://www.deepl.com/translator` - used in November 2021.

[5] DeepL forces the user to choose a variety of English (either British or American). This implies that the MT output could be expected to be (mostly) British English while the HT is a mix of both varieties. Hence, one could argue that variety is an aspect that could be picked up by the classifier. We also use Google Translate, which does not allow the user to select an English variety.

[6] We noticed that the free Python library *googletrans* had clearly inferior translations. The paid APIs for Google and DeepL obtain COMET (Rei et al., 2020) scores of 59.9 and 61.9, respectively, while the *googletrans* library obtains 21.0.

[7] If we apply a bit more fuzzy matching by only keeping ascii letters and numbers for each sentence, the percentages go up by around 0.5%.

Parameter	Range
Learning rate	$5 \times 10^{-6}, \mathbf{10^{-5}}, 3 \times 10^{-5}$
Batch size	$\{\mathbf{32}, 64\}$
Warmup	$\{\mathbf{0.06}\}$
Label smoothing	$\{\mathbf{0.0}, 0.1, 0.2\}$
Dropout	$\{0.0, \mathbf{0.1}\}$

Table 2: Hyperparameter range and final values (bold) for our final DEBERTA models. Hyperparameters not included are left at their default value.

3.2 Classifiers

SVM We will experiment with a number of different classifiers. As a baseline model, we use a linear SVM with unigrams and bigrams as features trained with `scikit-learn` (Pedregosa et al., 2011), for which the data is tokenized with `Spacy`.[8] The use of a SVM is mainly to find out how far we can get by just looking at the superficial lexical level. It also allows us to identify whether the classifier uses any shortcuts, i.e. features that are not necessarily indicative of a human or machine translation, but due to artifacts in the data sets, which can still be picked up as such by our models. An example of this is punctuation, which was mentioned in previous work (Bhardwaj et al., 2020). MT systems might normalize uncommon punctuation,[9] while human translators might opt for simply copying the originally specified punctuation in the source sentence (e.g. quotations, dashes). We analyse the importance of normalization in Section 5.

Fine-tuning LMs Second, we will experiment with fine-tuning pre-trained language models.[10] Fu and Nederhof (2021) only used BERT (Devlin et al., 2019b) and Bhardwaj et al. (2020) used a set of BERT- and ROBERTA-based LMs, but there exist newer pre-trained LMs that generally obtain better performance. We will empirically decide the best model for this task, by experimenting with a number of well-established LMs: BERT (Devlin et al., 2019b), RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2021b; He et al., 2021a), XLNet (Yang et al., 2019), BART (Lewis et al., 2020) and Longformer (Beltagy et al., 2020). For all these models, we only tune the batch size and learning rate. The

[8] `https://spacy.io/`

[9] The normalisation of the punctuation as a pre-processing step when training an MT system is a widespread technique, so that e.g. «, », ″, “ and „ are all converted to e.g. ″.

[10] Implemented using HuggingFace (Wolf et al., 2020).

		Acc.
BART-large	Lewis et al. (2020)	64.9
BERT-large	Devlin et al. (2019b)	61.9
DEBERTA-v3-large	He et al. (2021a)	68.6
Longformer-large	Beltagy et al. (2020)	63.5
ROBERTA-large	Liu et al. (2019)	65.5
XLNET-base	Yang et al. (2019)	62.3
DEBERTA-v3-large (optim)		68.9

Table 3: Best **development set** results (all in %) for MT vs HT classification for a number of pre-trained LMs. On the test set, DEBERTA-v3-large (optim) obtains an accuracy of 66.1.

best model from these experiments is then tuned further (on the dev set). We tune a single parameter at a time and do not perform a full grid search due to efficiency and environmental reasons. Hyperparameter settings and range of values experimented with are shown in Table 2.

Evaluation We evaluate the models looking at the accuracy and F1-score. When standard deviation is reported, we averaged over three runs. For brevity, we only report accuracy scores, as we found them to correlate highly with the F-scores. We include additional metrics, such as the F-scores, on our GitHub repository.

4 Experiments

SVM The SVM classifier was trained on the training set WMT08–17$_O$ (i.e. part of the data set with original source side), where the MT output was generated with DeepL. It obtained an accuracy of 57.8 on dev and 54.9 on the test set. This is in line with what would be expected: there is some signal at the lexical level, but other than that the task is quite difficult for a simple SVM classifier.

Finding the best LM As previously indicated, we experimented with a number of pre-trained LMs. For efficiency reasons, we perform these experiments with a subset of the training data (WMT14-17$_O$, i.e. with only translations from original text). The results are shown in Table 3. We find the best performance by using the DeBERTa-v3 model, which quite clearly outperformed the other LMs. We obtain a 6.7 point absolute increase in accuracy over BERT (61.9 to 68.6), the LM used by Fu and Nederhof (2021)), and a 3.7 point increase over the second best performing model, BART-large. We tune some of the remaining hyperparameters further (see Table 2) and obtain an accuracy of 68.9. We will use this model in our next experiments.

Trained on → ↓ Evaluated on	DeepL Acc.	Google Acc.
DeepL	66.1 ± 1.1	56.3 ± 0.3
Google	63.8 ± 1.6	64.9 ± 1.1
FAIR (Ng et al., 2019)	62.6 ± 1.9	57.7 ± 1.8
RWTH (Rosendahl et al., 2019)	61.9 ± 1.5	58.3 ± 1.8
PROMT (Molchanov, 2019)	50.3 ± 0.9	52.1 ± 3.3
online-X	57.5 ± 1.1	56.6 ± 3.4

Table 4: Test set scores (all in %) for training and testing our best DEBERTA across different MT-systems (DeepL and Google) and 4 WMT19 submissions. online-X refers to an anonymous online MT system evaluated at WMT19.

Cross-system performance A robust classifier that discriminates between HT and MT should not only recognize MT output that is produced by a particular MT system (the one the classifier is trained on), but should also work across different MT systems. Therefore, we test our DeepL-trained classifier on the translations of Google Translate (instead of DeepL) and vice versa. In this experiment we train the classifier on all the training data (i.e. WMT08-17$_{O+T}$) and evaluate on the test set.

In Table 4, we find that this cross-system evaluation leads to quite a drop in accuracy: 2.3% for DeepL and even 8.6% for Google. It seems that the classifier does not just pick up general features that discriminate between HTs and NMT outputs, but also MT-system specific features that do not always transfer to other MT systems.

In addition, we test both classifiers on a set of MT systems submitted to WMT19. We pick the two top and two bottom submissions according to the human evaluation (Barrault et al., 2019). The motivation is to find out how the classifiers perform on MT outputs of different levels of translation quality. We also notice a considerable drop in performance here. Interestingly, the classifiers perform best on the high-quality translations of FAIR and RWTH (81.6 and 81.5 human judgment scores at WMT19, respectively), and perform considerably worse on the two bottom-ranked WMT19 systems (71.8 and 69.7 human judgment scores). It seems that the classifier does not learn to recognize lower-quality MT outputs if it only saw higher-quality ones during training.

This inability to deal with lower-quality MT when trained only on high-quality MT seems counterintuitive and was quite surprising to us. After all, the difference between high-quality MT and human translation tends to be more subtle than in the case of low-quality MT. However,

	Dev	Test
WMT14-17$_{O+T}$	71.1 ± 1.3	64.9 ± 0.6
WMT14-17$_{O}$	68.9 ± 1.4	64.0 ± 1.1
WMT08-17$_{O+T}$	71.2 ± 0.9	66.1 ± 1.1
WMT08-17$_{O}$	71.5 ± 0.8	66.3 ± 0.5
WMT08-17$_{T}$	63.7 ± 0.8	59.5 ± 0.3

Table 5: Dev and test scores for training our best DEBERTA model on either WMT14-17 or WMT08-17 translated with DeepL, compared with training on the same data sets but not adding the *translationese data* (T) and only using T.

the learned features most useful for distinguishing high-quality MT from HT are likely different in nature than the features that are most useful for distinguishing low-quality MT from HT (e.g., simple lexical features versus features related to word ordering). From this perspective, feeding low-quality MT to a system trained on high-quality MT can be seen as an instance of out-of-distribution data that is not modelled well during the training stage. Nevertheless, this featural discrepancy could likely be resolved by supplying additional examples of low-quality MT to the classifier at training time.

Removing translationese data In our previous experiment we used the full training data (i.e. WMT08-17$_{O+T}$). However, most of the WMT data sets only consist for 50% of sentences that were originally written in German; the other half were originally written in English (see Section 3.1). We ask the question whether this additional data (which we refer to as *translationese*) is actually beneficial to the classifier. On the one hand, it is in fact a different category than human translations from original text. On the other, its usage allows us to double the amount of training data (see Table 1).

In Table 5 we show that the extra data helps if there is not much training data available (WMT14-17), but that this effect disappears once we increase the amount of training data (WMT08-17). In fact, the *translationese* data seems to be clearly of lower quality (for this task), since a model trained on only this data (WMT08-17$_{T}$), which is of the same size as the WMT08-17$_{O}$ experiments, results in quite a drop in accuracy (59.5 vs 66.3 on the test set). We have also experimented with pre-training on WMT08-17$_{O+T}$ and then fine-tuning on WMT08-17$_{O}$. Our initial results were mixed, but we plan on investigating this in future work.

Beyond sentence-level In many practical use-cases, we actually have access to full documents, and thus do not have to restrict ourselves to looking at just sentences. This could lead to better performance, since certain problems of NMT systems only come to light in a multi-sentence setting (Frankenberg-Garcia, 2021). Since WMT also contains document-level information, we can simply use the same data set as before. Due to the number of instances being very low at document level (see Table 1), and to the fact that the addition of *translationese* data showed to be beneficial with limited amounts of training data (see Table 5), we use all the data available for our document-level experiments, i.e. WMT08-17$_{O+T}$.

We have four document-level classifiers: (i) a SVM, similar to the one used in our sentence-level experiments, but for which each training instance is a document; (ii) majority voting atop our best sentence-level classifier, DEBERTA, i.e. we aggregate its sentence-level predictions for each document by taking the majority class; (iii) DEBERTA fine-tuned on the document-level data, truncated to 512 tokens; and (iv) Longformer (Beltagy et al., 2020) fine-tuned on the document-level data, as this LM was designed to handle documents.

For document-level training, we use gradient accumulation and mixed precision to avoid out-of-memory errors. Additionally, we truncate the input to 512 subword tokens for the DEBERTA model. For the dev and test set, this means discarding 11% and 2% of the tokens per document on average, respectively.[11] A potential approach for dealing with longer context without resorting to truncation is to use a sliding window strategy, which we aim to explore in future work.

The results are presented in Table 6. First, we observe that the document-level baselines obtain, as expected, better accuracies than their sentence-level counterparts (e.g. 60.7 vs 54.9 for SVM and 72.5 vs 66.1 for DEBERTA on test). Second, we observe large differences between dev and test, as well as large standard deviations. The instability of the results could be due, to some extent, to the low number of instances in these data sets (138 and 290, as shown in Table 1). Moreover, the test set is likely harder in general than the dev set, since it on average has fewer sentences per document (13.8 vs 21.7).

[11] The median subword token count in the HT document-level data is 376, with a minimum of 47 and maximum of 3,254.

	DeepL		Google	
	Dev	**Test**	**Dev**	**Test**
SVM	74.8	60.7	84.7	64.8
DEBERTA (mc)	84.7±8.0	72.5±5.2	93.2±1.1	67.6±3.4
DEBERTA	91.1 ± 2.4	76.8±4.4	95.9 ± 1.5	60.8±1.2
Longformer	80.2 ± 2.7	82.0±7.2	94.2±1.3	63.2±0.9

Table 6: Accuracies of training and evaluating on document-level DeepL and Google data. For DEBERTA, we try two versions: a sentence-level model applied to each sentence in a document followed by majority classification (mc), and a model trained on full documents (truncated to 512 tokens).

5 Discussion & Analysis

Thus far we have reported results in terms of an automatic evaluation metric: classification accuracy. Now we would like to delve deeper by conducting analyses that allow us to obtain further insights. To this end, we exploit the fact that the SVM classifier outputs the most discriminative features for each class: HT and MT.

5.1 Punctuation Normalization

In this first analysis we looked at the best features of the SVM to find out whether there is an obvious indication of "shortcuts" that the pre-trained language models can take. The best features for both HT and MT are shown in Table 8.

For comparison, we also show the best features after applying Moses' (Koehn et al., 2007) punctuation normalization,[12] which is commonly used as a preprocessing step when training MT systems. Indeed, there are punctuation-level features that by all accounts should not be indicative of either class, but still show up as such. The backtick (`) and dash symbol (–) show up as the best unigram features indicating HT, but are not present after the punctuation is normalized.

Now, to be clear, one might make a case of still including these features in HT vs MT experiments. After all, if this is how MT sentences can be spotted, why should we not consider them? On the other hand, the shortcuts that work for this particular data set and MT system (DeepL) might not work for texts in different domains or texts that are translated by different MT systems. Moreover, the shortcuts might obscure an analysis of the more interesting differences between human and machine translated texts.

[12] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/normalize-punctuation.perl

	Original	**Normalized**
Sent-level		
SVM	54.9	54.5
DEBERTA-v3	66.1 ± 1.1	67.0 ± 0.6
Doc-level		
SVM	60.7	60.0
DEBERTA (majority)	72.5 ± 5.2	72.0 ± 4.1
DEBERTA	76.8 ± 4.4	77.2 ± 4.7
Longformer	82.0 ± 7.2	83.7 ± 2.1

Table 7: Test set accuracies of training and evaluating on sentence-level and document-level data on either the original or normalized (by Moses) input texts, translated with DeepL.

In any case, we want to determine the impact of punctuation-level shortcuts by comparing the original scores versus the scores of our classifiers trained on punctuation-normalized texts. The results of our baseline and best sentence- and document-level systems with and without normalization are shown in Table 7. We observe that, even if the two best unigram features were initially punctuation, normalizing does not affect performance in a major way. There is even a small increase in performance for DEBERTA-v3 and Longformer, though likely not significant.

5.2 Unigram Analysis

In our second analysis we manually went through the data set to analyse the 10 most indicative unigram features for MT (before normalization).[13] Interestingly, some are due to errors by the human translator: the MT system correctly used *schoolyard* instead of the split *school yard*, and it also used the correct name *Olympiakos Piraeus* instead of the incorrect *Olypiacos Piraeus* (typo in the first word). Some are indeed due to a different (and likely better) lexical choice by the human translator, though the translation is not necessarily wrong: *competing gang* instead of *rival gang*, *espionage scandal* instead of *spy affair*, *judging panel* instead of *jury* and *radiation* instead of *rays*. Finally, the feature *disclosure* looks to be an error on the MT side. It occurs a number of times in the machine-translated version of a news article discussing Wikileaks, in which the human translator chose the correct *Wikileaks publication* instead of *Wikileaks disclosure* and *whistleblower activists* instead of *disclosure activists*.

[13] Of course, since we only look at unigrams here, and the performance of the sentence-level SVM is not very high anyway, all these features have in common that they do not necessarily generalize to other domains or MT-systems.

Before normalization				After normalization			
Most indicative for MT		**Most indicative for HT**		**Most indicative for MT**		**Most indicative for HT**	
1-grams	2-grams	1-grams	2-grams	1-grams	2-grams	1-grams	2-grams
olympiakos	are said	`	the riders	olympiakos	" proctor	u.s.	the riders
affair	" proctor	–	the 2015	affair	are said	program	consequently ,
forsa	2010 ,	u.s.	consequently ,	forsa	book "	nearly	the 2015
rival	per cent	nearly	projects ,	rays	2010 ,	anticipated	. the
rays	almost the	program	. the	rival	per cent	everybody	projects ,
schoolyard	the flat	anticipated	life "	disclosure	almost the	premier	<93>the hunting
disclosure	in view	<93>the	- weiss	jury	be put	lama <92>s	a part
jury	with industry	premier	a part	succeed	and later	weiss	as for

Table 8: Best features (1-gram and 2-gram models) in the SVM classifier per class, before and after normalizing punctuation.

For the best unigrams indicative of HT, there are some signs of simplification by the MT system. It never uses *nearly* or *anticipate*, instead generally opting for *almost* and *expected*. Similarly, human translators sometimes used *U.S.* to refer to the United States, while the MT system always uses *US*. The fact that we used British English for the DeepL translations might also play a role: *program* is indicative for HT since the MT system generally used *programme*.

6 Conclusions

In this paper we trained classifiers to automatically distinguish between human and machine translations for German-to-English. Our classifiers are built by pre-training state-of-the-art language models. We use the test sets of the WMT shared tasks, to ensure that the machine translation systems we use (DeepL and Google) did not see the data already during training. Throughout a number of experiments, we show that: (i) the task is quite challenging, as our best sentence-level systems obtain around 65% accuracy, (ii) using *translationese* data during training is only beneficial if there is limited data available, (iii) the accuracy drops considerably when performing cross MT-system evaluating, (iv) accuracy improves when performing the task on the document-level and (v) normalizing punctuation (and thus avoiding certain shortcuts) does not have an impact on model performance.

In future work, we aim to do a number of things. For one, we want to experiment with both translation directions and different source languages instead of just German. Second, we want to perform cross-domain experiments (as in Bhardwaj et al. (2020)), as we currently only looked at news texts.[14] Third, we want to look at the effect of the source language: does a monolingual model that is trained on English translations from German still work on translations into English from different source languages? This can shed on light on the question in what sense general source language-independent features that discriminate between HT and MT are actually identified by the model. Fourth, we plan to also use the source sentence, with a multilingual pre-trained LM, following Bhardwaj et al. (2020). This additional information is expected to lead to better results. While the source sentence is not always available, there are real-world cases in which it is, e.g. filtering crawled parallel corpora. Fifth, we would like to expand the task to a 3-way classification, as in the least restrictive scenario, given a text in a language, it could be either originally written in that language, human translated from another language or machine translated from another language.

7 Acknowledgements

The authors received funding from the European Union's Connecting Europe Facility 2014-2020 - CEF Telecom, under Grant Agreement No. INEA/CEF/ICT/A2020/2278341 (MaCoCu). This communication reflects only the authors' views. The Agency is not responsible for any use that may be made of the information it contains. We thank the Center for Information Technology of the University of Groningen for providing access to the Peregrine high performance computing cluster. Finally, we thank all our MaCoCu colleagues for their valuable feedback throughout the project.

[14] Note that this domain has a real-world application: the detection of fake news, given the fact that MT could be use to spread such news in other languages (Bonet-Jover, 2020).

References

Aharoni, Roee, Moshe Koppel, and Yoav Goldberg. 2014. Automatic detection of machine translated text and translation quality estimation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 289–295.

Ahrenberg, Lars. 2017. Comparing machine translation and human translation: A case study. In *Proceedings of the Workshop Human-Informed Translation and Interpreting Technology*, pages 21–28, Varna, Bulgaria, September. Association for Computational Linguistics, Shoumen, Bulgaria.

Arase, Yuki and Ming Zhou. 2013. Machine translation detection from monolingual web-text. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1597–1607.

Barrault, Loïc, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy, August. Association for Computational Linguistics.

Beltagy, Iz, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Bhardwaj, Shivendra, David Alfonso Hermelo, Phillippe Langlais, Gabriel Bernier-Colborne, Cyril Goutte, and Michel Simard. 2020. Human or neural translation? In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6553–6564, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Bonet-Jover, Alba. 2020. The disinformation battle: Linguistics and artificial intelligence join to beat it.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July. Association for Computational Linguistics.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. 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, June. Association for Computational Linguistics.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. 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, June. Association for Computational Linguistics.

Frankenberg-Garcia, Ana. 2021. Can a corpus-driven lexical analysis of human and machine translation unveil discourse features that set them apart? *Target. International Journal of Translation Studies*, 09.

Fu, Yingxue and Mark-Jan Nederhof. 2021. Automatic classification of human translation and machine translation: A study from the perspective of lexical diversity. In *Proceedings for the First Workshop on Modelling Translation: Translatology in the Digital Age*, pages 91–99, online, May. Association for Computational Linguistics.

He, Pengcheng, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

He, Pengcheng, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June.

Le, Hang, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2020. FlauBERT: Unsupervised language model pre-training for French. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France, May. European Language Resources Association.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July. Association for Computational Linguistics.

Li, Yitong, Rui Wang, and Hai Zhao. 2015. A machine learning method to distinguish machine translation from human translation. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation: Posters*, pages 354–360.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Martin, Louis, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online, July. Association for Computational Linguistics.

Molchanov, Alexander. 2019. Promt systems for wmt 2019 shared translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 302–307, Florence, Italy, August. Association for Computational Linguistics.

Ng, Nathan, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy, August. Association for Computational Linguistics.

Nguyen-Son, Hoang-Quoc, Tran Phuong Thao, Seira Hidano, and Shinsaku Kiyomoto. 2019a. Detecting machine-translated paragraphs by matching similar words. *arXiv preprint arXiv:1904.10641*.

Nguyen-Son, Hoang-Quoc, Tran Phuong Thao, Seira Hidano, and Shinsaku Kiyomoto. 2019b. Detecting machine-translated text using back translation. *arXiv preprint arXiv:1910.06558*.

Nguyen-Son, Hoang-Quoc, Tran Thao, Seira Hidano, Ishita Gupta, and Shinsaku Kiyomoto. 2021. Machine translated text detection through text similarity with round-trip translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5792–5797.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rei, Ricardo, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November. Association for Computational Linguistics.

Rosendahl, Jan, Christian Herold, Yunsu Kim, Miguel Graça, Weiyue Wang, Parnia Bahar, Yingbo Gao, and Hermann Ney. 2019. The RWTH Aachen University machine translation systems for WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 349–355, Florence, Italy, August. Association for Computational Linguistics.

Toral, Antonio and Víctor M. Sánchez-Cartagena. 2017. A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1063–1073, Valencia, Spain, April. Association for Computational Linguistics.

Toral, Antonio. 2019. Post-editese: an exacerbated translationese. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 273–281, Dublin, Ireland, August. European Association for Machine Translation.

Vanmassenhove, Eva, Dimitar Shterionov, and Andy Way. 2019. Lost in translation: Loss and decay of linguistic richness in machine translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 222–232, Dublin, Ireland, August. European Association for Machine Translation.

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

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.