Human or Neural Translation?

Shivendra Bhardwaj
David Alfonso-Hermelo
Philippe Langlais
RALI/DIRO
Université de Montréal
Montreal (Quebec) H3C 3J7, Canada
felipe@iro.umontreal.ca

Gabriel Bernier-Colborne
Michel Simard
Cyril Goutte
National Research Council Canada
1200 Montreal Road, Building M-58
Ottawa, Ontario K1A 0R6, Canada
Cyril.Goutte@nrc-cnrc.gc.ca

Abstract

Deep neural models tremendously improved machine translation. In this context, we investigate whether distinguishing machine from human translations is still feasible. We trained and applied 18 classifiers under two settings: a monolingual task, in which the classifier only looks at the (French) translation; and a bilingual task, in which the source text (in English) is also taken into consideration. We report on extensive experiments involving 4 neural MT systems (Google Translate, DeepL, as well as two systems we trained) and varying the domain of texts. We show that the bilingual task is the easiest one and that transfer-based deep-learning classifiers perform best, with mean accuracies around 85% in-domain and 75% out-of-domain.

1 Introduction

This work addresses the task of distinguishing between translations produced by humans and machines. Practical applications for this include: improving machine translation systems (Li et al., 2015), filtering parallel data mined from the Web (Arase and Zhou, 2013) and evaluating machine translation quality without reference translations (Aharoni et al., 2014). In our case, we are more interested in tracing the origin of translations outsourced by a large institutional translation service.

Our work aims at distinguishing between human and neural machine translations at the sentence level. We consider two settings: a monolingual task, where only the target sentence is considered; and a bilingual task where both the source text and its translation are available. We compare feature-based approaches with several deep learning methods, investigating the impact of text domains and MT systems (in-house neural engines, Google Translate, DeepL), paying attention to cases where the translation engine at test time is different from the one used for training, which we found often not studied in related work. We show that identifying machine translation is still feasible nowadays. On the bilingual task, the best transfer learning method we tested recorded an in-domain accuracy of 87.6% and out-of-domain performances ranging between 65.4% and 84.2% depending on the domain of texts and MT system considered. We analyze why our classifiers manage to do better than chance even though translations produced automatically seem to us of very good quality overall. We believe our study offers many new data points, and hope it will foster research on this timely topic.

After reviewing related work in Section 2, we describe our dataset and experimental setting in Section 3, the neural MT systems we used in our experiments in Section 4 and the classifiers we tested in Section 5. We present our experimental results in Section 6 and propose a deeper analysis in Section 7.

2 Related Work

Most studies on identifying machine translation were conducted at a time where MT systems were fraught with problems that rendered their identification somewhat easy. Current neural MT systems deliver translations that are sometimes bafflingly fluent. We are not aware of much work addressing MT identification with these newer systems. One notable exception is a recent study by Nguyen-Son et al.

Proceedings of the 28th International Conference on Computational Linguistics, pages 6553–6564
Barcelona, Spain (Online), December 8-13, 2020

(2019) on distinguishing original sentences from translations produced by Google Translate (GT). The authors build on the interesting intuition that back translations of automatically translated texts should contain less variations (word usage, structure) than back translations of human translations. They report an accuracy of 75% with an SVM classifier on a small corpus of 1200 sentences selected from the Europarl corpus[1] that are either original (human) or translated with GT. In their experiments, they use the same translation engine for producing the automatic translation of test sentences, and the back-translations used by the classifier. In a real-world scenario, we are not expected to know which system has been used for producing a translation (we do not even know if a translation has been produced automatically) and the impact of producing back-translations by a different translation engine remains to investigate.

In earlier work on MT identification, approaches and evaluations vary greatly from one study to the other. For instance, Li et al. (2015) uses features extracted from the parse tree of the sentence to characterize, as well as features capturing the density of some function words (with the help of a part-of-speech tagger), and some features dedicated to out-of-vocabulary words. They also use features aimed at capturing emotion agreement inside a sentence, using a dictionary of emotion words. They gathered a balanced dataset of human and machine translations from the Europarl corpus (including French-English, German-English, Italian-English and Danish-English language pairs) using a statistical machine translation (SMT) engine trained in-domain with Moses (Koehn et al., 2007). They report an accuracy of 74.2%. However, they do not analyze which features are the most beneficial to the task.

Arase and Zhou (2013) investigate the use of features to capture the fluency of the text, such as part-of-speech and word-based n-gram language models, as well as features aimed at detecting so-called *phrase-salad* phenomena (Lopez, 2008), i.e. poor inter-phrasal coherence often observed in SMT output. On a collection of public texts crawled over the Web, they report an impressive accuracy of 95.8% when distinguishing human versus automatic translations for the English-Japanese language pair. The best performance was observed when combining all the features, and surpasses that of humans performing the same task (88.2%). The authors did not show the quality of the automatic translations, but mentioned that it was pretty low compared to the translations produced by native speakers and professional translators. It is therefore questionable how their approach would do on good quality automatic translations.

Aharoni et al. (2014) use features capturing the presence or absence of part-of-speech tags and function words taken from LIWC (Pennebaker et al., 1999) appearing at least 10 times in the training material. On a corpus extracted from the Canadian Hansards, and using various translation engines, they report accuracies at detecting machine versus human translations (under a monolingual scenario tested on the English language) which are inversely correlated with the quality of the MT system used. For the best translation engines, they report an accuracy slightly over 60%.

3 Data

All our experiments are centered around one very large dataset: the translation memory of a large institutional translation service. This data collection — called TM hereafter — contains the English and French versions of over 1.8 million documents, covering over 200 broad domains (military, health, etc.), for a total close to 140 million sentence pairs. Since the vast majority of translations in the TM are into French, we focus on this language direction.

Our goal is to build classifiers that determine if a translation is human or machine-made. For this, we need training data that contains both types of translations. We create such data by machine translating a subset of 530k sentence pairs, randomly sampled from the TM. These machine translations are performed using two different neural MT systems, themselves trained using a distinct subset of 5.8M sentence pairs, also randomly sampled from the TM.[2] These two MT systems, one based on XLM (Lample and Conneau, 2019) and one on FairSeq (Ott et al., 2018), are detailed in Section 4. Thus, two distinct classifier training sets are created, one from each MT system: each contains 530k human translations and 530k machine translations, totalling 1.06M examples.

[1] `http://www.statmt.org/europarl`

[2] All sampling in the TM was done in such a way as to ensure comparable representations of each domain.

We proceed similarly to produce test sets to evaluate the performance of our classifiers: we randomly sample 10k sentence pairs from the TM, machine translate the English versions into French using our XLM and FairSeq MT systems, thus creating two test sets of 20k examples (10k human translations + 10k machine translations) each. We call these X-TM (for XLM) and F-TM (for FairSeq).

These two test sets can be seen as "in-domain" relative to our classifiers: not only because they share the same source as the training data (the TM), but also because the machine translations were produced using the same MT systems. To test the ability of our classifiers to handle different text domains and translations produced by different MT engines, we also created "out-of-domain" test sets: we used two online translation platforms — DeepL[3] (D) and Google Translate[4] (GT) — to translate 10K sentences of each of four publicly available data sets: Europarl (EURO), Canadian Hansard (HANS),[5] the News Commentaries (NEWS) available through the WMT conference,[6] and the Common Crawl corpus (CRAWL) also available through WMT. Again these were mixed in equal parts with human translations. In what follows, each test set is named based on the system used to produce automatic translations, and the domain of the material.

We further translated another excerpt of (previously unused) 10k sentences from the TM, using the *DeepL* translation API with a private account, to produce a test set we call D-TM. The TM being a proprietary translation memory, we did not submit it to the GT platform.

4 NMT systems

As noted above, to produce the training data for our classifiers, we first created two transformer-based NMT systems using English-French texts from the TM. We provide the details of this process here.

4.1 Cross-lingual Language Model (XLM)

In Lample and Conneau (2019), the authors propose three models: two unsupervised ones that do not use sentence pairs in translation relation, and a supervised one that does. We focus on the third model, called the Translation Language Modeling (TLM) which tackles cross-lingual pre-training in a way similar to the BERT model (Devlin et al., 2018a) with notable differences. First, XLM is based on a shared source-target vocabulary using Byte Pair Encoding (BPE) (Sennrich et al., 2016). We used the 60k BPE vocabulary which comes with the pre-trained language model.[7] Second, XLM is trained to predict both source and target masked words, leveraging both source and target contexts, encouraging the model to align the source and target representations. Third, XLM stores the ID for the language and the token order (*i.e.*, positional encoding) in both languages which builds a relationship between related tokens in the two languages.

During training and when translating, we use a beam search of width 6 and a length penalty of 1. XLM is implemented in PyTorch[8] and supports distributed training on multiple GPUs.[9] The original distribution does not include beam search for translating (but does for training), so we modified it accordingly. Also, we modified the pre-processing code such that XLM accepts a parallel corpus for training TLM.

4.2 Scaling Neural Machine Translation (FairSeq)

Scaling NMT (Ott et al., 2018) is a novel transformer model that showcased an improvement in training efficiency while maintaining state-of-the-art accuracy by lowering the precision of computations, increasing the batch size and enhancing the learning rate regimen. The architecture uses the `big-transformer` model with 6 blocks in encoder and decoder networks. The half-precision training reduced the training time by 65%. Scaling NMT is implemented in PyTorch and is part of the `fairseq-py` toolkit.[10] We use the default 40k vocabulary with a shared source and target BPE factor-

[3] `www.deepl.com/translator`
[4] `https://translate.google.com/`
[5] `https://www.isi.edu/natural-language/download/hansard/`
[6] `https://www.statmt.org/wmt14/translation-task.html`
[7] The model without pre-training was unstable. We noticed better results with a long-running back-translation step.
[8] `https://pytorch.org/`
[9] `https://github.com/facebookresearch/XLM.git`
[10] `https://github.com/pytorch/fairseq`.

ization. During training and for translating, we use a beam search of width 4 and a length penalty of 0.6. For translation,[11] we average the last five checkpoints.

4.3 Post-processing

Translating the classifier training data (Section 3) with the XLM engine took approximatively 10 hours on a computer equipped with a V100-SXM2 GPU, and 26 hours for the FairSeq system. By inspection, we noticed small issues with the translations produced by both systems, such as punctuation misplacements, extra spaces, inconsistencies in the use of single and double-quotes. Since those issues would ease the identification of machine-translated material, we normalized the translations in a post-processing step, using 12 very conservative regular expressions[12] that we applied to both the human and machine translations. We observe in Table 1 a clear increase of BLEU when applying normalization: +4 for XLM, and +5.3 for FairSeq.

	raw	normalized
XLM	33.43	37.46
FairSeq	34.07	39.40

Table 1: BLEU scores of the XLM and FairSeq translation engines measured on a dataset of 550K sentence pairs (described in Section 3) before (left) and after (right) normalization,

5 MT Identification

We experimented with two strategies for building classifiers: feature-based models trained from scratch, as well as deep learning ones making use of pre-trained representations.

5.1 Feature-based Classifiers

We considered three supervised classifiers informed by different feature sets. We tested various classifiers (random forest, support vector machines and logistic regression), but obtained more stable results with random forest classifiers trained with `scikit-learn` (Pedregosa et al., 2011). In all our experiments, we fixed the number of trees in the forest to 1000 with a maximum depth of 40 and a minimum number of samples required to split an internal node set to 10.

n-GRAM We reproduce the approach of Cavnar and Trenkle (1994) where we define a vector space on the 30k most frequent character n-grams in the MT output of our training material, with n ranging from 2 to 7.[13] Each sentence is then encoded by the frequency of the terms in this vocabulary, thus leading to a large sparse representation which is passed to a classifier. In the bilingual task, we also consider the top 30k n-grams of the source-language version of the training corpus, leading to representations of 60k dimensions.

KENLM As a point of comparison, in the monolingual task, we experimented with features extracted from four $\{3, 4\}$-gram word language models trained with the `kenLM` package (Heafield et al., 2013) on the machine-translated material of our training corpus: two left-to-right models, and two right-to-left ones. We computed 18 features: ratios of min and max `logprob` over the (target) sentence per model (four features), the number of tokens with a `logprob` less than $\{mean, max, -6\}$ (three features per model), as well as the `logprob` of the full sentence given by the left-to-right models (two features).

T-MOP T-MOP (Jalili Sabet et al., 2016) is a translation memory cleaning tool which computes 27 features for detecting spurious sentence pairs, including broad features (such as length ratio) adapted from Barbu (2015), some based on IBM models computed by MGIZA++ (Gao and Vogel, 2008), as well

[11] We used the `fairseq-interactive` module of the `fairseq-py` toolkit[10].

[12] Very specific rules such as `replace(' ;',';')` or `replace('https :','https:')`.

[13] Larger vocabularies do not yield notable performance differences.

as some features based on multilingual word embeddings, using the method proposed by Søgaard et al. (2015). While in T-MOP, those features are aggregated in an unsupervised way (that is, with rules), we instead pass them to a random forest classifier trained specifically to distinguish human from machine translations. Because of the nature of the feature set, we only deploy this classifier in the bilingual task.

5.2 Deep Learning Classifiers

bi-LSTM We re-implemented the method of Grégoire and Langlais (2018) for recognizing whether two sentences are translations of each other: two bidirectional LSTMs (Hochreiter and Schmidhuber, 1997) encode the source and target sentences into two continuous vector representations, which are then fed into a Feed-Forward Neural Network with two layers (one in the original paper): one of dimension 150 to process the continuous representation, and one of dimension 75. The output of each network is finally passed to the sigmoid function.

In the original paper, the authors used 512-dimensional word embeddings and 512-dimensional recurrent states since they learn the word embedding from scratch. We found it easier (faster, and slightly better) to adapt pre-trained FAST word embeddings (Bojanowski et al., 2016) of dimension 300. Also, the authors tie the parameters of the two encoders, while we do not. We use two hidden layers before the sigmoid function because we are mapping from 300 values to 1 and intuitively, it is better to do it smoothly. We trained our classifier with the Adadelta optimizer (Zeiler, 2012) with gradient clipping (clip value 5) to avoid exploding gradient and batch size 300, whereas the original architecture uses the Adam optimizer with a learning rate of 0.0002 and a mini-batch of 128.[14]

We use a similar setting for the monolingual task, except that we only use one bidirectional LSTM whose output we directly pass to the hidden layer of dimension 150, then a layer of dimension 75 and finally the sigmoid function.

LASER The LASER toolkit (Artetxe and Schwenk, 2019) released by Facebook[15] provides a pre-trained sentence encoder that handles 92 different languages. Sentences from all the languages are mapped together into the same embedding space with a bi-LSTM 512-dimensional encoder, such that the embeddings from different languages are comparable.

For the bilingual detection task, we extract the representation of the source and target sentences and tie them into one vector by taking their absolute difference and dot product, and adding them. This tied representation is then passed through 3 hidden layers of size 512, 150 and 75 respectively[16] with dropout (Srivastava et al., 2014) of 50%, and then fed into a relu (Nair and Hinton, 2010) activation function, whose output is finally passed to the sigmoid function. For the monolingual task, we just use the LASER French (target) representation of the sentence and pass it through the very same architecture. We train the classifiers with the Adadelta optimizer with gradient clipping (clip value 3).

Transformer-based Classifiers The use of pre-trained language models in a transfer learning setting is ubiquitous and has shown substantial improvements in various NLP tasks. Therefore, we also considered various representations trained either solely on French data (CamemBERT, FlauBERT) or on multiple languages (XLM-ROBERTA, XLM, and mBERT). We experiment with different pre-trained transformer models, using the Python module `simpletransformers`[17] based on the `HuggingFace` library[18], which has a sequence classification head on top (a linear layer on top of the pooled output). Our classifiers were fine-tuned using the `ClassificationModel` class and evaluated with the `eval_model` class. We have maintained the same parameters for all the transformer models: sequence length of 256, batch size of 32, Adam optimizer (Kingma and Ba, 2014)[19].

[14] Adadelta does not require to set a default learning rate, since it takes the ratio of the running average of the previous time-steps to the current gradient.

[15] `https://github.com/facebookresearch/LASER`.

[16] We used three layers here because the input dimension is larger (512 versus 300).

[17] `https://github.com/ThilinaRajapakse/simpletransformers`

[18] `https://github.com/huggingface/transformers`

[19] lr: $1 \times e^{-5}$, adam_epsilon: $1 \times e^{-8}$

	TM			NEWS		CRAWL		HANS		EURO	
	X-	F-	D-	GT-	D-	GT-	D-	GT-	D-	GT-	D-
Feature-based classifiers:											
n-GRAM	76.0	76.6	81.4	66.6	72.6	59.2	61.9	47.2	49.6	53.6	56.3
KENLM	80.2	80.4	58.6	49.8	49.6	50	49.6	49.1	49.7	50.3	50.2
Deep-learning classifiers:											
bi-LSTM	64.5	62.7	53.3	60.8	59.3	57.7	55.7	**57.9**	55.5	58.5	57.4
LASER	55.9	56.3	58.4	54.8	54.5	54.5	53.9	54.7	50.5	54.1	53.6
Transformer-based classifiers:											
CamemBERT	83.7	**83.8**	73.8	68.9	**77.3**	**63.0**	**68.8**	52.3	**58.5**	56.6	60.5
XLM-ROBERTA	83.0	83.5	75.1	67.4	76.6	60.1	66.5	51.2	58.0	55.2	60.0
FlauBERT	**84.3**	82.2	**82.4**	**71.3**	77.0	64.8	66.4	51.7	53.8	**59.8**	**61.5**
XLM	79.9	77.5	72.3	69.8	73.2	60.3	61.0	49.9	50.1	54.9	56.0
mBERT	78.4	78.8	72.2	70.9	74.4	60.5	61.5	49.4	50.2	54.8	56.0

Table 2: Accuracy of classifiers on the monolingual classification task, on all test sets. X, F, D, and GT refer to the XLM, FairSeq, DeepL, and Google translation engines, respectively. Underlined scores are produced by classifiers trained with XLM material; FairSeq material has been used otherwise.

CamemBERT (Martin et al., 2019) is based on the RoBERTa (Liu et al., 2019) architecture (which is basically a BERT model with improved hyper-parameters for robust performance) and is trained on 138GB of plain French text taken from multilingual corpus OSCAR (Ortiz Suárez et al., 2019). Unlike RoBERTa, CamemBERT uses sentence piece tokenization (Kudo and Richardson, 2018) and performs whole word masking, which has been shown to be preferable (Joshi et al., 2019). The architecture of the base model is a multi-layer bidirectional transformer (Devlin et al., 2018b; Vaswani et al., 2017) with 12 transformer blocks of hidden size 768 and 12 self attention heads.

FlauBERT (Le et al., 2019) The base model we used is trained on 71GB of publicly available French data and the data was pre-processed and tokenized using a basic French tokenizer (Koehn et al., 2007). The model was trained with the MLM training objective.

XLM-ROBERTA (Ruder et al., 2019) is a multilingual language model, trained on 100 different languages. It is an extended version of XLM (see Section 4.1).

mBERT (Devlin et al., 2018b) is very similar to the original BERT model with 12 layers of bidirectional transformers, but released as a single language model trained on 104 separate languages from Wikipedia pages, with a shared word piece vocabulary. The model does not use any marker for input language and the pre-trained model is not made to extract translation pairs to have similar representations. The tokenization splits words into multiple pieces and it takes the prediction of the first piece as the prediction for the word. The model is fine-tuned to minimize cross-entropy loss.

6 Experiments

We trained all classifiers described above using training data produced with XLM and FairSeq MT systems. Overall, classifiers trained with FairSeq translations performed very marginally better on out-of-domain data, with an average accuracy of 64.5%, compared to 64.3% for classifiers trained with XLM translations. In this section, unless otherwise specified, we report the results of classifiers trained with FairSeq translations, but both training sets produce very comparable results.

6.1 Monolingual task

Results on the monolingual task are reported in Table 2. Most accuracies are over the 50% that would be obtained by a random guess, albeit by a small margin on some conditions. Expectedly, the best performances are observed on in-domain data (TM), in which machine translations were produced by the same MT systems used to produce the classifiers' training data. Which of XLM or FairSeq was used to produce test translations has little to no impact on performance, however. The highest accuracy (84.3%)

	TM			NEWS		CRAWL		HANS		EURO	
	X-	F-	D-	GT-	D-	GT-	D-	GT-	D-	GT-	D-
Feature-based classifiers:											
n-GRAM	76.2	77.9	81.9	66.8	73.2	59.2	62.1	54.4	51.8	49.6	47.2
T-MOP	62.7	62.9	59.8	63.4	62.9	61.1	57.2	54.8	57.8	51.2	50.1
Deep-learning classifiers:											
bi-LSTM	66.5	65.2	57.8	68.9	65.8	71.6	68.7	65.5	63.6	66.6	57.0
LASER	68.0	68.8	68.3	77.2	75.1	80.8	78.5	**73.5**	50.3	73.2	63.1
Transformer-based classifiers:											
CamemBERT	**87.5**	**87.6**	**84.6**	76.3	84.2	77.8	82.2	66.8	**73.1**	71.3	65.4
XLM-ROBERTA	86.7	85.8	81.2	76.2	82.5	77.5	79.7	67.2	68.5	69.8	63.3
FlauBERT	84.9	84.1	81.8	76.3	81.7	75.4	75.4	61.7	62.9	68.9	62.7
XLM	84.3	82.4	83.5	75.5	79.5	76.5	77.1	58.0	58.7	64.0	55.8
mBERT	86.6	83.9	82.9	**81.1**	**85.7**	**83.2**	**83.1**	70.6	58.3	**76.8**	**68.3**

Table 3: Accuracy of classifiers on the bilingual classification task, on all test sets. Underlined scores are produced by classifiers trained with XLM material; FairSeq material has been used otherwise.

is obtained on TM data by fine-tuning the FlauBERT pre-trained representations on the training material produced with XLM. Using this configuration, but classifying translations produced by DeepL only slightly reduces performance (82.4%), but for most other approaches — including other BERT-inspired solutions — it does lead to a notable decrease of accuracy.

HANS and EURO are the hardest test sets, where performances are often close to the random guess baseline. This suggests that translations produced by GT and DeepL on those datasets are very good and hard to distinguish from human translations. Part of this poor performance may be imputed to some extent to the mismatch between the system used to translate the classifiers's training material, and the one used for testing. The lowest performances overall are recorded when classifying sentences produced by GT on the HANS dataset, where the best classifier only succeeds at a rate of 57.9%. Around 15% of automatic translations in this test set are identical to the reference one (see column 1 of Table 4), and the same percentage are very close to the reference (see column 2 and 3). It is notorious that GT has been trained on Hansards, further complicating the task. We were however surprised by the low percentage of automatic translations close to the reference one we measured on the EURO test set. Inspection did not reveal anything particular. If we set apart those two test sets, we observe that BERT-like models provide better results than bi-LSTM and LASER ones. BERT models are systematically better at classifying DeepL translations than those produced by GT. We do not have a clear explanation for this yet.

The n-GRAM feature-based classifier is competitive with the LASER and bi-LSTM classifiers, but is slightly behind BERT-inspired classifiers. KENLM is clearly overfitting, delivering impressive results for such a simple device on in-domain data and systems, but failing to generalize to other settings.

The good performances we obtained on TM, when distinguishing translations produced by DeepL may be of interest to the language service provider that provided us with the data. It could for instance be used to diagnose translation providers that heavily rely on this system to produce their translations. The performance obtained on the NEWS and CRAWL test sets indicate that the automatic translations do have a signature that we can recognize to some extent, without even looking at the source sentence.

6.2 Bilingual Task

Table 3 shows accuracies obtained in the bilingual task, that is, when both the source sentence and the translation are considered. With a very few exceptions, all configurations benefit the extra input. For settings where the monolingual accuracies are high, the gains can be modest (for instance less than 2 points for FlauBERT on in-domain test sets), but otherwise, clear improvements are observable. For instance, on the HANS test sets, gains close to 20 points can be observed for some Transformer-based classifiers.

The more challenging datasets are now handled with an accuracy around 70% or above, while for the other test sets, the best performances are over 80%. Similarly to the monolingual task, Transformer-

	=ref %	1-edit %	2-edit %	BLEU	Monolingual Task		Bilingual Task	
F-TM	6.1	2.4	1.3	39.3	83.8	(CAM)	87.6	(CAM)
X-TM	5.3	1.9	1.1	37.8	<u>84.3</u>	(FLAU)	<u>87.5</u>	(CAM)
GT-EURO	3.5	0.4	0.4	37.4	59.8	(FLAU)	76.8	(MBERT)
D-TM	4.8	5.9	2.3	36.2	82.4	(FLAU)	84.6	(CAM)
GT-HANS	15.5	9.8	1.5	34.9	57.9	(LSTM)	73.5	(LASER)
D-HANS	14.1	11.9	2.6	34.6	58.5	(CAM)	73.1	(CAM)
D-NEWS	1.8	0.6	0.2	33.4	77.3	(CAM)	85.7	(MBERT)
GT-NEWS	1.8	0.5	0.3	32.0	71.3	(FLAU)	81.1	(MBERT)
D-EURO	2.0	0.6	0.3	31.8	61.5	(FLAU)	68.3	(MBERT)
GT-CRAWL	1.5	0.7	0.3	25.2	63.0	(CAM)	83.2	(MBERT)
D-CRAWL	1.5	0.6	0.2	25.0	68.8	(CAM)	83.1	(MBERT)

Table 4: Accuracy of best classifier (in percentage) for each test set, in the monolingual and bilingual tasks, as a function of the (normalized) BLEU score. Except for underlined scores, classifier training data were produced with XLM. The best classifier is specified in parentheses next to its accuracy. Column "=ref %" indicates the percentage of sentences for which MT output is identical to the reference; while "x-edit%" columns indicate the percentage of translations which differe to the reference translation by exactly one or two edit distance operations.

based classifiers are the best performers. The T-MOP classifier overall underperforms the `bi-LSTM` and LASER ones. The n-GRAM classifier shows signs of overfitting, and delivers disappointing results on out-of-domain data.

7 Analysis

7.1 Quantitative Analysis

Table 4 shows the accuracy of the best performing classifiers for each test set, alongside the BLEU score of the respective translation engine (F-, X-, GT-, D-) for that set. We anticipated that poor quality MT would be easier to detect, but BLEU score does not seem to correlate strongly with the classification performance, which contradicts the observation in Aharoni et al. (2014). What is noticeable however, is that in-domain performances (data from TM, and classifiers trained with the same translation engine used for producing test-sentence translations) are systematically higher than out-domain ones. Also, the bilingual task is unquestionably easier to tackle and for many test sets, including out-of-domain ones, the best classifier achieves an accuracy over 80%, a rather decent level of performance we did not anticipate at first, considering the relatively high quality of current NMT output.

Figure 1 shows the cumulative accuracy (y-axis) in the bilingual task calculated over the number of target sentences, sorted by the length of sentences (number of tokens). For all test sets and all classifiers, we observe that the longer the translation, the better the accuracy. This corroborates the findings of Arase and Zhou (2013), that longer sentences are easier to classify. This is likely explained by the fact that translations of short sentences are more likely to be similar to the human translation, and longer sentences likely contain more problems, further easing detection.

7.2 Qualitative Analysis

We inspected the decisions made by our classifiers on some examples. We did notice machine translations involving problems with proper names and acronyms, as example i) of Figure 2. We also occasionally found syntax problems in machine translations, such as example ii), which involves a failure in long-distance number agreement as well as a bad choice of pronoun. Also, we observed a strong tendency of machine translations to mimic the structure of the source sentence, as can be seen in most examples of Figure 2. This suggests that alignment features in the bilingual task could be useful. T-MOP explicitly captures alignment information, but does not seem to make good use of it. We were otherwise impressed

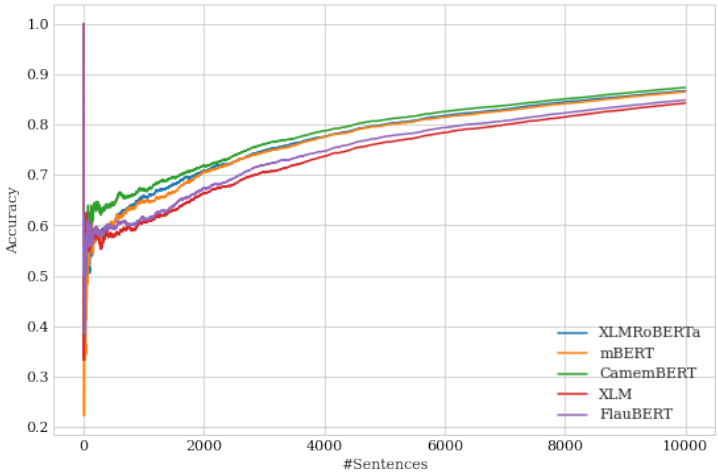

Figure 1: Cumulative accuracy in the bilingual task calculated over the number of target sentences produced by the XLM engine, sorted by their number of tokens.

by the overall quality of the MT, and rapidly realized how difficult it would be for human annotators to achieve a decent level of performance on this task. This is in line with the observations of Arase and Zhou (2013), who report lower performances for humans than for machines at detecting translations produced by statistical phrase-based MT.

To better understand the type of information our classifiers base their decisions on, we inspected cases where our classifiers predominantly classified the human translations as such[20], and the machine translation counterpart is predominantly recognized as a machine translation. For 32 such cases randomly selected, we manually produced minimal pairs (3 on average), that is, as small as possible variants of the automatic translation, to see at which point the classifiers were changing their decisions from machine to human, thus allowing us to see which signals they react to. For instance, we produced 7 variants of example i) in Figure 2, including the 3 reported.

We found that in half of the cases, modifying only a few words (often only one) of the automatic translation is enough for the classifier to reverse its decision. Some cases involved normalizations that our post-processing script (see Section 4.3) fails to take into account. Among those, we noted the presence of a hyphen symbol produced by DeepL on the NEWS data set, different from the one used in human translations. We also noted a few cases involving typographical preferences. For instance, on the EURO test set, removing a space in section numbering produced by XLM (*e.g.* "5 c)" versus "5c)") sometimes suffices to make our classifiers believe the translation is human. Also, removing a capital letter (or sometimes adding one) may reverse the classifier's decision.

Of course, such normalization issues are in a way deceptive since although they do help decision making, they do not have much to do with translation quality. In any case, the most frequent situation involves lexical choices. For instance in example v) of Figure 2, changing the future tense *enverra* by the infinitive form *doit envoyer* significantly reduces (from 18 to 4) the number of classifiers believing the translation is an automatic one. Further replacing the preposition *pour* by *en vue de* reduces this number to 2. Sometimes, it is easy to blame the translation engine for a different lexical choice, as the underlined wording in example iii), but sometimes it is less, as in example iv) where *se retire* might be a correct translation of *step down*. Clearly, more analysis is required to better appreciate the type of information captured by our classifiers.

8 Conclusion

In this study, we implemented 18 classifiers to detect machine-translated texts, and evaluated their performance on several test sets, containing translations produced by different state-of-the-art NMT systems.

[20]By "predominant", we mean that at least 15 out of our 18 classifiers agreed.

i) SRC 6c) Were you informed about the **ADR** process at the CHRC?
HUM 6c) Vous a-t-on informé du processus relatif au **RAD** de la CCDP ?
NMT 6c) Avez-vous été informé du processus de **MARC** à la CCDP ? (XLM,TM)

VAR 6c) Avez-vous été informé du processus de *RAD* à la CCDP ? (14)
VAR 6c) *Vous a-t-on informé* du processus de MARC *de* la CCDP ? (9)
VAR 6c) *Vous a-t-on informé* du processus *au RAD de* la CCDP ? (8)

ii) SRC Are there any specific services being requested by SMEs that you are not able to provide for them or that you feel lie outside of your mandate?
HUM Les PME vous demandent-elles de leur fournir des services que vous ne pouvez leur donner ou qui, selon vous, échappent à votre mandat
NMT Y a-t-il des services particuliers demandés par les PME que vous ne pouvez pas leur fournir ou **que**, selon vous, ne **cadre** pas avec votre mandat ? (XLM, TM)

iii) SRC Until 2004, my parents met Nhan Thi Duong my ex-girlfriend and **asked for my daughter** Lan Thu Thi Le.
HUM Ils n'ont rencontré Nhan Thi Duong, mon ex-petite amie, qu'en 2004, et lui ont demandé **des nouvelles de ma fille**, Lan Thu Thi Le.
NMT Jusqu'en 2004, mes parents ont rencontré Nhan Thi Duong, mon ex-petite amie, et m'ont demandé **de me donner ma fille** Lan Thu Thi Le. (XLM, TM)

iv) SRC A bigger bloodbath seems inescapable if he does not step down.
HUM Il semble difficile d'échapper à un bain de sang plus important encore s'il n'accepte pas de démissionner.
NMT Un plus grand bain de sang semble inévitable s'il ne se retire pas. (DeepL, NEWS)

v) SRC Action Baki to send reminder for October inspection.
HUM Mesure Baki doit envoyer un rappel en vue de l'inspection d'octobre.
NMT Mesure Baki enverra un rappel pour l'inspection d'octobre. (XLM, TM)

VAR Mesure Baki *doit envoyer* un rappel pour l'inspection d'octobre. (4)
VAR Mesure Baki *doit envoyer* un rappel *en vue de* l'inspection d'octobre. (2)

Figure 2: Examples of human and automatic translations. Underlined passages identify problems and bold ones their corresponding parts. Examples i) and v) come along manually edited variants of the automatic translation (modifications are in italic), followed in parentheses by the number of classifiers (among 18) that identify an automatic translation.

Overall, we found that classifiers with access to both the source sentence and the translation perform better than those with access to the translation alone. Our classifiers achieve accuracies above 80% on several test sets and always surpass a random baseline. Our analysis reveals that, despite of our efforts to normalize translations, artifacts still exist in the data that could explain in part our relatively high classifier accuracies. But in general, it appears that NMT systems do elicit signatures that can be recognized by automatic methods. Often, a single lexical choice gives away the automatic nature of the translation, even when the translation looks fluent from a language model point of view.

While we had the opportunity to work on a large, high quality professional translation memory, we realize that our results can not be *replicated* exactly: by nature, large professional TMs are proprietary and not easily shared. We argue however that one can easily *reproduce* (Drummond, 2009) our experiments in another setting.

In future work, we hope to produce better MT detectors by creating training data using a wider variety of MT systems. Another question we would like to examine is to what extent it is possible to detect post-edited translations, i.e. machine translations manually edited by human translators.

Acknowledgements

We acknowledge the support of the Translation Bureau through a research collaboration with the RALI/Université de Montréal and the National Research Council of Canada.

References

Roee Aharoni, 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, Baltimore, Maryland, June. Association for Computational Linguistics.

Yuki Arase 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, Sofia, Bulgaria, August. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Trans. Assoc. Comput. Linguistics*, 7:597–610.

Eduard Barbu. 2015. Spotting false translation segments in translation memories. In *Proceedings of the Workshop Natural Language Processing for Translation Memories*, pages 9–16.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

W.B. Cavnar and J.M. Trenkle. 1994. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018a. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Chris Drummond. 2009. Replicability is not reproducibility: nor is it good science. In *Proceedings of the Evaluation Methods for Machine Learning Workshop at the 26th ICML*.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, Ohio, June. Association for Computational Linguistics.

Francis Grégoire and Philippe Langlais. 2018. Extracting parallel sentences with bidirectional recurrent neural networks to improve machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1442–1453, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria, August. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

Masoud Jalili Sabet, Matteo Negri, Marco Turchi, José G. C. de Souza, and Marcello Federico. 2016. TMop: a tool for unsupervised translation memory cleaning. In *Proceedings of ACL-2016 System Demonstrations*, pages 49–54, Berlin, Germany, August.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Philipp Koehn, 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 *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual Language Model Pretraining. *arXiv e-prints*, page arXiv:1901.07291, Jan.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2019. FlauBERT: Unsupervised Language Model Pretraining for French.

Yitong Li, Rui Wang, and Hai Zhao. 2015. A machine learning method to distinguish machine translation from human translation. In *PACLIC*.

Yinhan Liu, 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. *CoRR*, abs/1907.11692.

Adam Lopez. 2008. Statistical machine translation. *ACM Computing Surveys (CSUR)*, 40(3).

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2019. CamemBERT: a Tasty French Language Model.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814.

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

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*, Cardiff, United Kingdom, July.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. *CoRR*, abs/1806.00187.

F. Pedregosa, 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.

James Pennebaker, Martha Francis, and Roger Booth. 1999. Linguistic inquiry and word count (liwc). 01.

Sebastian Ruder, Anders Søgaard, and Ivan Vulić. 2019. Unsupervised cross-lingual representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 31–38, Florence, Italy, July. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. 2015. Inverted indexing for cross-lingual nlp. In *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015)*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Matthew D. Zeiler. 2012. Adadelta: An adaptive learning rate method.