Evaluating Translation Quality Using Transformer Models

Xabier de Zuazo Oteiza
University of the Basque Country
(UPV/EHU)
xzuazo002@ikasle.ehu.eus

Carlos Domínguez Becerril
University of the Basque Country
(UPV/EHU)
cdominguez019@ikasle.ehu.eus

Abstract

Human evaluations of machine translation are extensive but expensive. These evaluations can take months to finish and involve human labor that can not be reused. Moreover, the overall machine translation quality available for professional translators working with Spanish - Basque pairs is relatively poor and scarce. In this paper, we propose several methods that, given a machine-translated sentence in Basque from Spanish, the system obtains the Translation Error Rate (TER) metric without giving any reference sentence. We present this method as an alternative to evaluating the quality of machine translation models without any reference sentence, human supervision, or relying on other features as the input of our system is going to be only the machine-translated text. As the results show, metrics predicted by fine-tuned transformer models have a considerable correlation with the evaluations made by humans.

1 Introduction

Human evaluations of Machine Translation (MT) are challenging to perform as they weigh many aspects of translations, such as adequacy, fidelity, or fluency (Hovy, 1999). Besides, these human evaluation approaches are expensive, can take weeks or months to complete, and can not be reused. In addition, human evaluations are a big problem for researchers and developers of machine translation systems. This is because they need to test the effect of daily changes on their systems in order to improve them.

One possible alternative to human evaluations is automatic metrics. The automatic metrics tend to be inexpensive, fast, language-independent, and highly correlated with human evaluations. The most used metrics are BLEU (Papineni et al., 2002), NIST (Doddington et al., 2000), TER (Snover et al., 2006), chrF (Popović, 2015), and COMET (Rei et al., 2020), to name a few. The problem with using automatic metrics is that language is ambiguous, and the structure is complicated. Therefore, estimating whether a translation is correct or to what extent it can be considered correct is a lot more laborious. Two entirely different sequences of words (sentences) may be completely equivalent, while two sequences that differ in small details can have entirely different meanings. Furthermore, the performance measured automatically on a benchmark may not carry over to a different body of text, especially in a different domain.

The work on this paper focuses on evaluating machine-translated sentences in Basque without a reference sentence by predicting the TER metric. The dataset used is the one provided by (Aranberri and Pascual, 2018) which contains Basque machine translations from Spanish along with some extra features. In this research, we propose to use a fine-tuned IXAmBERT transformer (Otegi et al., 2020) and design several systems. Additionally, for comparison, we will work with BERTeus (Agerri et al., 2020) and other Basque-based transformer models too. The main objective of our project is to use only the machine-translated text as an input, as it is the natural way of evaluation for humans; that is, we will not rely on extra features.

2 Related Work

Some researchers have already explored the possibility of using Deep Learning (DL) and Machine Learning (ML) techniques to accomplish the requirements of an efficient TER metric evaluator. Under the big umbrella of artificial intelligence, both ML and DL aim to learn meaningful information from the given enormous language resources. However, the techniques differ in the methodologies to learn valuable information from the corpus. On the one hand, the ML-based TER metric evaluator relies primarily on feature engineering. On the other hand, DL-based systems are better at learning complex features from raw data without relying on feature engineering because of their deep structure.

In recent years, there have been different approaches for automated evaluation of machine translation metrics with and without reference sentences using DL. For example, some researchers (Shimanaka et al., 2019) propose to use a BERT transformer (Devlin et al., 2018) by inserting the machine translation sentence along with the reference sentence to obtain the Direct Assessment (DA) human evaluation scores metric. The DA measures to what extent a hypothesis adequately expresses the meaning of the reference translation, calculating the mean of a large number of human evaluations for each translation.

Other researchers (Aranberri and Pascual, 2018) propose to extract a set of 17 baseline features using the Quest++ (Specia et al., 2015) in order to train several ML models and measure the TER metric. These features are black-box features, that is, shallow MT system-independent features. Most of them rely on comparing the sentences against a large training corpus, e.g., language model probabilities, n-gram frequencies, and translation options per word.

In this project, we will use the IXAmBERT transformer (Otegi et al., 2020). IXAmBERT is a multilingual language pre-trained transformer for English, Spanish and Basque. The training corpora is composed of the English, Spanish, and Basque Wikipedias, together with Basque crawled news articles from online newspapers. The model has been successfully used to transfer knowledge from English to Basque in a conversational QA system. We will also use the BERTeus transformer, a monolingual model for Basque, which has been trained on a corpus comprising Basque online crawled news articles and the Basque Wikipedia.

3 Dataset Description

This study makes use of the dataset provided by (Aranberri and Pascual, 2018). This dataset was adapted from a workshop ran in 2015, which contained post-editing data collected from professional translators. The dataset contains sentences in Spanish and the machine translation of the sentences to Basque. Additionally, it provides the human edited version of the machine-translated text, several features extracted from the text using Quest++ (Specia et al., 2015), and other extra information, such as the edit types and their frequencies, and the time needed to fix the machine-translated text by a human evaluator.

The original dataset comes with a train and a test partition. Here the 20% of the original training set has been used to create a development split. Table 1 shows the final contents of each split.

Split	Sentences	Tokens
Train	4397	95,615
Dev	1100	23,880
Test	712	14,886
Total	6209	134,381

Table 1: Distribution of the data across splits.

In Figure 1, we can see the distribution of the TER values in the dataset. Most of the scores in the dataset are between 0 and 100, and the values are generally quite balanced between the splits. Another thing to take into account is how there is a peak of 0 that is more prominent in the test set. These values represent translations accepted as correct or with minimal changes. With the intention to alleviate this small imbalance, augmentation has been carried out in the training split. This augmentation consists of adding instances with the same target and post-edit values and a TER score of zero. Besides, we expected this to help the model better understand which solutions are reasonable and need no modification. The dataset has been augmented only by a 10% to avoid accidentally inserting a bias towards low values.

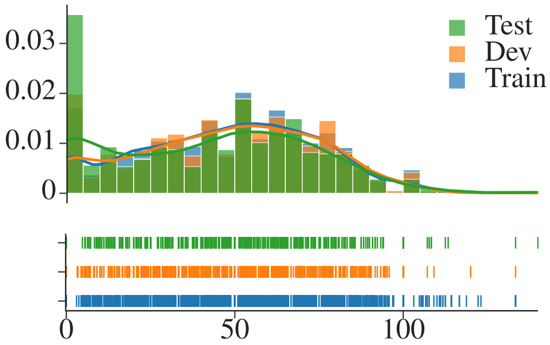

Figure 1: Dataset TER values density plot.

4 Models

The models used in our experiments are BERT-based models with Basque language support. More specifically, we will use IXAmBERT, BERTeus, and different RoBERTa models. Figure 2 shows the architecture of the model. We put a multilayer perceptron (MLP) on the [CLS] token embedding with a final layer of one neuron to adapt the model for the regression task.

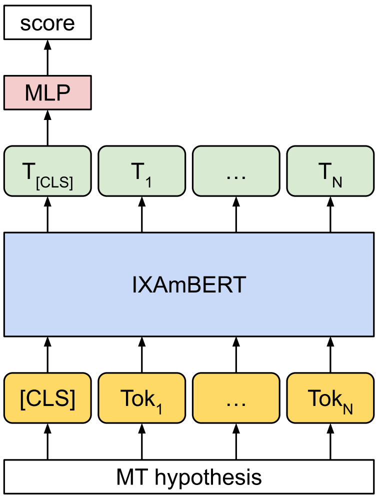

Figure 2: IXAmBERT/BERTeus model architecture for regression.

Unfortunately, when working with transformer encoders for regression, there is an instability issue in the output results when performing the fine-tuning. It is a difficulty documented in multiple articles (Peters et al., 2019; Merchant et al., 2020) and for which no solution has yet been found. This complication is because these models have not been originally trained for this kind of task. Until now, the best-known approach to get some stable results is to use the original Adam BERT optimizer, try different initialization seeds and methods, and train for more epochs (Dodge et al., 2020; Zhang et al., 2020; Mosbach et al., 2020). In our case, training for more epochs only seem to cause over-fitting of the model, but trying different initialization methods and seeds helps greatly to find models that are able to learn and progress correctly. At the same time, the seeds make the results reproducible.

Consequently, the models here have been fine-tuned with multiple seeds each and trying different hyper-parameters. Then, the average and best model scores were extracted. With the aim of searching for a good correlation between TER and our predicted score, the RMSE loss was used during the training. The specific hyperparameters of the model are the following:

- Seeds $\in \{0, 1, 2, 3, 4\}$
- Number of epochs $\in \{1, 2, 3, 4, 5\}$
- Learning rate (Standard Adam) $= 3e - 4$
- Batch size $= 6$
- Dropout rate for MLP layer $1 = 0.1$
- The number of hidden layers on the MLP $= 3$
- The number of hidden units of the MLP $\in \{768, 512, 256\}$
- Weight initialization: Xavier Uniform (Glorot and Bengio, 2010)

5 Experiments

In this project, five experiments are proposed using different amount of features. Nonetheless, all of them will share the same approach: the text will be the main source to obtain the TER score. In other words, contrary to the techniques previously used to predict the TER (Aranberri and Pascual, 2018), here, we will try not to depend on extra features.

The names of the experiments subsections refer to the following characteristics:

- **Spanish**: The source text of the translation.
- **Basque**: The machine-translated text.
- **Basque Post-Edit**: The post-edit corrections of the machine-translated text by a human evaluator.
- **Extra Features**: The edit types and their frequencies after correcting the machine translated text (post-edited text).

5.1 Experiment 1: Basque

The first experiment consists in using the Basque machine-translated text as input. That is to say, the raw text will be inserted in the transformer, and the TER metric is going to be obtained by regression.

5.2 Experiment 2: Spanish + Basque

We used the source text in Spanish and the machine-translated text in Basque in this second experiment. In particular, both raw texts will be sent to the model input splitting them by the [SEP] sequence separator.

5.3 Experiment 3: Basque + Basque Post-Edit

In the third experiment, we insert the Basque machine-translated text into the transformer together with the Basque post-edited text with a probability of 50% during training and 0% during testing. In this experiment, we want to do a teacher forcing-like experiment (Williams and Zipser, 1989), where we help the system by adding information that only is available during training and then removing it during testing. We think this extra information can be helpful so that the system can compare both sentences.

5.4 Experiment 4: Basque + Extra Features

The fourth experiment is similar to the third one. Nevertheless, in this case, we insert the Basque machine-translated text into the system and the edit types frequencies with a probability of 50% during training. During testing, as we do not have the frequency of the edit types, we insert -1s instead, as if to say that the feature is not available (in training, the other 50% of the time is trained by also inserting -1s). Similarly, as in the previous experiment, we think that this extra information can be helpful because the system may detect where the errors are.

5.5 Experiment 5: Basque + Post-Edit + Extra Features

The last experiment is a combination of the previous ones. In this case, we insert the Basque machine-translated text and, with a 50% probability, the post-edited Basque text, and the edit types frequencies. Like in the previous experiments, we want to know whether adding extra information during the training can help to improve the results in the testing split.

6 Evaluation

Evaluation of the task is not easy. Different metrics are usually used to estimate the quality of the TER score obtained. In this paper, we will make use of four metrics: Pearson's correlation, Spearman's correlation, root mean square error, and mean absolute error. Particularly, we will look closely at the Pearson's correlation, as it is the most widespread metric used when evaluating this task.

- **Pearson's correlation**: The Pearson correlation measures the strength of the linear relationship between two variables. It has a value between -1 to 1, with a value of -1 meaning a total negative linear correlation, 0 being no correlation, and +1 meaning a total positive correlation.

- **Spearman's correlation**: Spearman's correlation is a non-parametric measure of rank correlation. It assesses how well the relationship between two variables can be described using a monotonic function. If there are no repeated data values, a perfect Spearman correlation of +1 or -1 occurs when each variable is a perfect monotone function of the other.

- **Root Mean Squared Error (RMSE)**: Measures the differences between values (sample or population values) predicted by a model or an estimator and the values observed. The RMSE represents the square root of the second sample moment of the differences between predicted and observed values or the quadratic mean of these differences. RMSE is always non-negative, and a value of 0 would indicate a perfect fit for the data.

- **Mean Absolute Error (MAE)**: Measures the differences between values predicted by a model or an estimator and the values observed. The MAE represents the absolute differences between predicted values and observed values. MAE is always non-negative, and a value of 0 would indicate a perfect fit for the data.

7 Results

In Table 2, we can see the average scores of some BERT models with Basque support in Experiment 1 (see Section 5.1). Each model has been trained with five different seeds to calculate their mean and standard deviation. From the table can be concluded that the most appropriate models for this task are BERTeus and IXAmBERT, as they outperform the other models on all the metrics.

In Tables 3 and 4, we can find the maximum scores of IXAmBERT and BERTeus, respectively, for Experiments 1 to 5 (Sections 5.1 to 5.5). Experiments 1 and 2 with text input only have the best Pearson scores on both models, leading to the conclusion that adding the extracted features and post-edited version does not help the models improve. Additionally, including the source text in Spanish got a little worse results in BERTeus. This may mean that post-edit corrections to the translation are mostly related to the target language, not

Model	RMSE		MAE		Pearson		Spearman	
	$\mu_r \downarrow$	$\delta_r \downarrow$	$\mu_m \downarrow$	$\delta_m \downarrow$	$\mu_p \uparrow$	$\delta_p \downarrow$	$\mu_s \uparrow$	$\delta_s \downarrow$
ixambert	**25.06**	**0.93**	**19.34**	**1.04**	**0.5864**	**0.0281**	**0.5989**	**0.0236**
berteus	**24.60**	**0.54**	**18.81**	**0.46**	**0.6123**	**0.0191**	**0.6246**	**0.0132**
roberta-eus-cc100	27.28	1.98	21.46	2.06	0.5303	0.0700	0.5513	0.0809
roberta-eus-mc4	27.81	1.47	22.05	1.59	0.5005	0.0535	0.5345	0.0495
roberta-eus-euscrawl	27.39	2.32	21.92	2.70	0.4959	0.1527	0.5151	0.1618
roberta-eus-euscrawl-large	29.02	1.55	24.09	1.97	0.3369	0.2056	0.3486	0.2194

Table 2: Average scores in the test set of BERT models with Basque support (Experiment 1).

Model	RMSE↓	MAE↓	Pearson↑	Spear.↑
Basque	24.02	18.14	0.6226	**0.6300**
Spanish+Bas	23.85	18.62	**0.6501**	0.6511
Bas+post-edit	24.46	18.31	0.6364	0.6480
Bas+features	**23.68**	**17.38**	0.6367	0.6403
Bas+pedit+feat	25.54	19.42	0.6168	0.6321

Table 3: IXAmBERT best model scores in the test set (Experiments 1 to 5). "Bas" refers to Basque, "pedit" to "post-edit" and "feat" to features.

Model	RMSE↓	MAE↓	Pearson↑	Spear.↑
Basque	23.84	19.59	**0.6735**	**0.6542**
Spanish*+Bas	24.54	19.29	0.6343	0.6441
Bas+post-edit	23.85	**18.01**	0.6399	0.6482
Bas+features	23.82	19.17	0.6447	0.6365
Bas+pedit+feat	**23.59**	18.81	0.6576	0.6670

Table 4: BERTeus best model scores in the test set (Experiments 1 to 5).
*Note: BERTeus has not been originally pre-trained to support Spanish.

to the translation from the source language intrinsically. However, one thing to keep in mind is that BERTeus is monolingual and has not been pre-trained with texts in Spanish. That is not the case for the IXAmBERT model, for which including the source in Spanish does imply some improvement but still did not outperform the monolingual model.

In Figure 3, we show the predictions on the test set of the best model sorted by the gold score TER value. The plotted Human values represent the average TER value from different annotators for each particular segment. There is no doubt that there is room for improvement, particularly on translations requiring more post-edit changes and higher TER scores. However, the model is learning to differentiate which translations are better, and its predictions achieve a high correlation, comparable with average human performance.

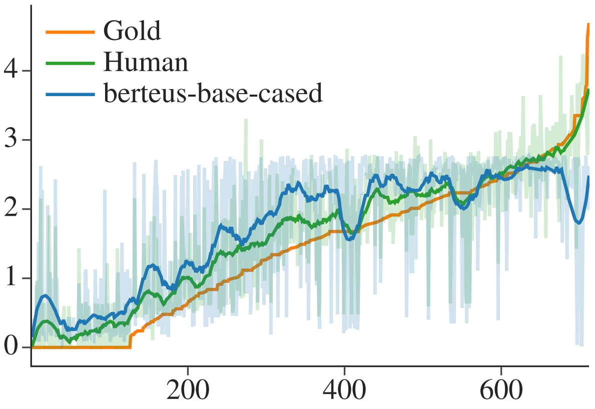

Figure 3: Predicted TER values in test set, normalized, and in ascending order.

Table 5 contains a summary of our best results compared to others' work. Considering that BERT-based models have not been previously pre-trained for regression tasks, they still get pretty close scores to the state-of-the-art models. The best transformer-based model is BERTeus, with a maximum Pearson correlation of 67%. Indeed, the standard deviation in this model is lower than the best machine learning model and even humans, which means that the metrics predicted are pretty stable.

8 Conclusions and Future Work

This study proposed using transformer-based models for the TER metric prediction. More specifically, these models have been used in a regression task set up to learn to evaluate machine learning translations. Even though the scores obtained are still not surprisingly good, the models show unmistakable signs of learning to evaluate translations correctly. The results show a high correlation, near 70%, between the human estimations and the scores predicted by the models. Another conclusion to be drawn from the results is that monolingual models also perform well on this type of task.

	Best Model			Pearson	
Model	RMSE↓	MAE↓	Pearson↑	$\mu_p \uparrow$	$\delta_p \downarrow$
Human TER	-	-	0.86	0.8243	0.0209
LR (Aranberri and Pascual, 2018)	-	-	-	0.3499	0.0399
k-NN (Aranberri and Pascual, 2018)	-	-	-	**0.7146**	**0.0220**
CART (Aranberri and Pascual, 2018)	-	-	-	0.6704	0.0367
SVR (Aranberri and Pascual, 2018)	-	-	-	0.3211	0.0415
MLP (Aranberri and Pascual, 2018)	-	-	-	0.4704	0.0517
ixambert + Basque	24.02	18.14	0.62	0.5865	0.0281
ixambert + Basque + Spanish	23.85	18.62	**0.65**	**0.5997**	**0.0187**
ixambert + Basque + post-edit	24.46	18.31	0.64	0.5764	0.0287
ixambert + Basque + features	**23.68**	**17.38**	0.64	0.5819	0.0252
ixambert + Basque + post-edit + features	25.54	19.42	0.62	0.5688	0.0320
berteus + Basque	23.84	19.59	**0.67**	**0.6123**	**0.0191**
berteus + Basque + Spanish	24.54	19.29	0.63	0.5656	0.0712
berteus + Basque + post-edit	23.85	**18.01**	0.64	0.5936	0.0302
berteus + Basque + features	23.82	19.17	0.64	0.6017	0.0201
berteus + Basque + post-edit + features	**23.59**	18.81	0.66	0.6004	0.0410
roberta-eus-cc100 + Basque	24.80	18.34	0.61	0.5303	0.0700
roberta-eus-mc4 + Basque	24.48	19.52	0.58	0.5005	0.0535
roberta-eus-euscrawl + Basque	23.42	19.32	**0.67**	0.4959	0.1527
roberta-eus-euscrawl-large + Basque	26.20	19.69	0.58	0.3369	0.2056

Table 5: Best Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Pearson's Correlation error on the test dataset using different models. We also provide the Pearson's mean and standard deviation between the five seeds.

After getting the first indications that these transformer-based models are able to progress in learning, many doors are opened to further the research. For future work, other pre-trained models could be tried to tackle the TER prediction task, like multi-language models or different MLP architectures, continue doing hyper-parameter tuning, data augmentation techniques, and others. Adding punctuation-related features may also help, as these models' tokenizers ignore all punctuation. Last but not least, speaking more broadly, it might be interesting to pre-train transformer-based models with multiple regression tasks to ameliorate their stability.

References

Rodrigo Agerri, Iñaki San Vicente, Jon Ander Campos, Ander Barrena, Xabier Saralegi, Aitor Soroa, and Eneko Agirre. 2020. Give your text representation models some love: the case for basque. *CoRR*, abs/2004.00033.

Nora Aranberri and Jose A Pascual. 2018. Towards a post-editing recommendation system for spanish–basque machine translation.

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

George R Doddington, Mark A Przybocki, Alvin F Martin, and Douglas A Reynolds. 2000. The nist speaker recognition evaluation–overview, methodology, systems, results, perspective. *Speech communication*, 31(2-3):225–254.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*.

Eduard H Hovy. 1999. Toward finely differentiated evaluation metrics for machine translation. In *Pro-*

ceedings of the EAGLES Workshop on Standards and Evaluation Pisa, Italy, 1999.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? *CoRR*, abs/2004.14448.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines. *CoRR*, abs/2006.04884.

Arantxa Otegi, Aitor Agirre, Jon Ander Campos, Aitor Soroa, and Eneko Agirre. 2020. Conversational question answering in low resource scenarios: A dataset and case study for basque. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 436–442.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. *CoRR*, abs/1903.05987.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. *arXiv preprint arXiv:2009.09025*.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2019. Machine translation evaluation with BERT regressor. *CoRR*, abs/1907.12679.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with quest++. In *Proceedings of ACL-IJCNLP 2015 system demonstrations*, pages 115–120.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. 2020. Revisiting few-sample BERT fine-tuning. *CoRR*, abs/2006.05987.