A regression approach on the Most Recent Metrics for Machine Translation

Carolina Neves, Filipe Coelho, Nguyen Huy Phuc
m20200049, m20200580, m20200566

1 Introduction

Natural Language Processing (NLP) is a growing technology that transforms the unstructured text in documents and databases to structured data for analysis; with the goal to qualify the computes to understand natural language.

This project will focus on a segment of NLP, the Machine Translation, which is a field that investigates the use of algorithms capable of translating text or speech from one language to another. To create a good capable automated translation the software is evaluated through metrics. These metrics correlate the translation proposed by a machine and the human assessment of quality of the translation. To simplify, the candidate translation refers to the machine's translation and the human-generated translations are the reference translations.

Given a certain corpus, with the original segment, the reference translation and with a normalized human quality assessment score, the purpose of this project is to create a metric that correlates with human assessments of quality.

2 Method/Approach

Corpus and Preprocessing

The corpus consists of six different language pairs, including the languages: Russian, English, German, Czech, Chinese, and Finish. A simple pre-processing was applied, to achieve an accurate metric, able to translate different languages. From the text it was only kept the letters, from a to z, and, applied lowercasing. However, for the Chinese translation, from English to Chinese, the *jieba* library was used to separate the different words.

ROUGE

ROUGE includes a set of different metrics, the ones explored in this project are recall, precision and f1 score. The recall counts the number of overlapping n-grams found in both the candidate model and reference, dividing then this number by the total number of n-grams in the reference. It allows to understand if the model is capturing all the information in the reference, however it is not good to identify if the model is just gathering a huge number of words with no meaning. Precision, in other hand, can tackle this issue by using a slightly different calculation, it considers the number of overlapping n-grams found in both the candidate and reference, and divides it by the total number of n-grams in the model n-gram count. The f1 score simply considers the previous metrics and calculates the following formula 2*((precision * recall) / (precision + recall)). This gives a more reliable measure of the model performance that relies not only on the model capturing as many words as possible – recall-, but also capturing only the most relevant words - precision.

A drawback of ROUGE is that it only evaluates at a syntactical level, not being able to measure at a semantic level, thus, it cannot accommodate for different words that have the same meaning. Due to this consequence, when providing different sentences using different words but with the same meaning, it will assign a low score.

BLUE

BLUE is a metric that computes the precision by calculating the fraction of tokens from the candidate translation that appear on the references. The return value is a number between 0, indicating a bad score, and 1, indicating the best score. An important feature to consider is that BLUE penalizes words that appear in the candidate more times than it appears in the references, which allows to retrieve a worse precision score if the candidate only includes repetitive words. However, it does not consider the order by which the words appear.

In practical terms, three functions were created to apply the BLUE metric. Firstly, *fromGroups* allows to apply the model to n-grams, to understand if the performance improves when it considers more than one word at a time. The function *BLUE* includes the preprocessing on the references and on the candidate translations, then it computes the word frequencies for each type of translation, and finally the score is calculated by computing the coverage of each word. The last function is called to apply of the corpus.

BLUERT

A more robust metric is the BLUERT which builds upon recent advances in transfer learning to capture widespread linguistic phenomena, such as paraphrasing. This metric can capture non-trivial semantic similarities sentences, being able to consider different sentences that convey the same meaning. It has as its basis a trained public collection of ratings (the WMT Metrics Shared Task dataset). Furthermore, it uses the contextual word representations of BERT and other metrics and models from the Machine Translation literature, combined with the pre-training scheme to increase BLEURT's robustness.

Evaluation

To use as baseline to train and evaluate the model, the given corpus had a column with the z-scores. Taking into account this reference, three metric scores were considered to check the performance of the translation metrics, the mean squared error, the pearson correlation, and the kendall correlation. The higher the score obtained in each evaluation metric the better the model is performing, since it is getting closer to the original score provided by the human assessment.

	mse	pcorr	kendallcorr
bleu_w1	0.898179	0.512997	0.341524
bleu_w2	1.065216	0.418989	0.293123
rouge_recall_w1	0.925880	0.497407	0.329099
rouge_precision_w1	0.881087	0.522616	0.348401
rouge_f1_w1	0.888932	0.518201	0.341341
rouge_precision_w2	1.055880	0.424243	0.296875
rouge_recall_w2	1.081815	0.409647	0.288378
rouge_f1_w2	1.062589	0.420468	0.292348
BLEURT	0.870491	0.528580	0.335507

Figure 1 – Evaluation of Scores Metrics (*en-fi*)

Final Model

The scores from 9 metrics obtained in the evaluation step were normalised using the standard scaler. The scores were then treated as input features for a regression model. Gradient Boosting (GB) is the model used for this task. As a highly overfitting nature of the GB model, a cross validation approach was implemented, a resampling procedure to evaluate the model, using a k-fold of 5 splits. The final evaluation score is the correlation between the out-of-fold predicted values and "z-score" column.

A grid search was applied to understand which parameters worked the best to each model, firstly to the Gradient Boosting Regressor model by scikit-learn and, then, to the GB model from LightGBM library.

```
'learning_rate': [0.01,0.03,0.05,0.09],
 'subsample'    : [0.9, 0.8,0.7],
 'n_estimators' : [100,200,300,800],
 'max_depth'    : [4,6,8,10],
learning_rate': [0.01,0.03,0.05,0.09],
'subsample'    : [0.9, 0.8,0.7],
'n_estimators' : [100,200,300,800],
'max_depth'    : [4,6,8,10],
```

Figure 2 – GridSearchCV hyperparameters space for the two models

Test Classification

Finally, after the model was trained on the train dataset, it was applied on the test dataset to evaluate the model on unseen data. It followed the same pipeline applied to the train dataset, first the pre-processing and, then, application of the best metrics found for each pair of languages. The result is a set of the metrics predictions.

3 Results and Discussion

Results

3.1.1. Preprocessing and Scoring metrics

As expected, after applying each metric to the corpus, BLEURT was the one to achieve the best scores for most translations.

The results of the 9 metrics were compared between the above-mentioned preprocessing steps and without preprocessing applied. Although, most pairs of language performed better with preprocessing, the pairs ***ru-en*** and ***en-fi*** performed better without preprocessing. Thus, it is decided that these two language pairs will be proceeded without preprocessing applied.

LP	Pre-processing	Best metric	Pearson Corr.	Kendall Corr.
cs-en	Yes	BLEURT	0.46526	0.31908
de-en	Yes	BLEURT	0.37235	0.25533
en-fi	**No**	BLEURT	0.52858	0.33550
en-zh	Yes	rouge_f1_w1	0.43608	0.29550
ru-en	**No**	BLEURT	0.38951	0.27466
zh-en	Yes	BLEURT	0.36078	0.22784

3.1.2. Regression model

For each pair of languages, the best model was applied, below it is possible to see which model and metrics delivered the best results in the trained dataset:

LP	Metrics & Model	Out-off-fold Pearson Corr. with "z-score"
cs-en	Best metrics: BLEURT	0.4652
	Best regression model: GB	0.5260
de-en	Best metrics: BLEURT	0.3723
	Best regression model: GB	0.4038
en-fi	Best metrics: BLEURT	0.5285
	Best regression model: LGBM	0.5374
en-zh	Best metrics: BLEURT	0.4360
	Best regression model: LGBM	0.4457
ru-en	Best metrics: BLEURT	0.3895
	Best regression model: LGBM	0.4116
zh-en	Best metrics: BLEURT	0.3607
	Best regression model: LGBM	0.3904

As can be seen, the result from the regression model significantly improved the correlation score (Pearson) for 6 language pairs

Test Performance

The performance of the test dataset will be evaluated by the teacher performing the correlation of the results obtained with the "z-score".

4 Discussion

There are several advanced metrics have not been accessed in this project such as BERT or COMET. The pre-trained embedding models such as LaBSE also not been used due to high computational expensive and short time of the project. In the future, the scores can be significantly improved if the mentioned approaches can be applied.

5 Conclusion

It is believed that the objective of the project was achieved with the implementation of a model for each pair of languages given in the corpus. Although, further improvements could be applied, for example, with the implementation of different and more recent metrics, the results were good and it was possible to test on the given test dataset.

Acknowledgments

References

Shmueli, B. (2021). NLP Metrics Made Simple: The BLEU score. Medium. Retrieved 21 May 2021, from https://towardsdatascience.com/nlp-metrics-made-simple-the-bleu-score-b06b14fbdbc1.

Briggs, J. (2021). The Ultimate Performance Metric in NLP. Medium. Retrieved 21 May 2021, from https://towardsdatascience.com/the-ultimate-performance-metric-in-nlp-111df6c64460.

Sellam, T., & Parikh, A. (2020). Evaluating Natural Language Generation with BLEURT. Google AI Blog. Retrieved 21 May 2021, from https://ai.googleblog.com/2020/05/evaluating-natural-language-generation.html#:~:text=BLEURT%20is%20a%20novel,%20machine,ratings%20provided%20by%20the%20user.

Appendices

1. SOURCE CODE

The code and metrics used in this project is available in our Github repository:
https://github.com/NOVA-IMS-20200580/TextMining

2. Correlation scores of 9 the metrics used for each language pair

- CS-EN:

With preprocessing

	mse	pcorr	kendallcorr
bleu_w1	1.030644	0.41819	0.277034
bleu_w2	1.054337	0.404586	0.280812
rouge_recall_w1	1.082813	0.388236	0.258786
rouge_precision_w1	0.990877	0.441024	0.296214
rouge_f1_w1	1.001665	0.434829	0.290947
rouge_precision_w2	1.065749	0.398033	0.27559
rouge_recall_w2	1.113455	0.370641	0.254624
rouge_f1_w2	1.07743	0.391326	0.267828
BLEURT	0.948657	0.465266	0.31908

Without preprocessing

	mse	pcorr	kendallcorr
bleu_w1	1.053754	0.404921	0.271061
bleu_w2	1.079262	0.390275	0.272864
rouge_recall_w1	1.11344	0.37065	0.24905
rouge_precision_w1	1.019766	0.424436	0.2866
rouge_f1_w1	1.035963	0.415136	0.279233
rouge_precision_w2	1.08908	0.384637	0.267882
rouge_recall_w2	1.135497	0.357986	0.247705
rouge_f1_w2	1.100779	0.37792	0.260021
BLEURT	0.940529	0.469932	0.329338

- DE-EN:

With preprocessing

	mse	pcorr	kendallcorr
bleu_w1	1.235359	0.289943	0.197475
bleu_w2	1.226609	0.29506	0.209101
rouge_recall_w1	1.244662	0.284504	0.19789
rouge_precision_w1	1.207247	0.306381	0.213228
rouge_f1_w1	1.201046	0.310007	0.214917
rouge_precision_w2	1.228761	0.293801	0.206706
rouge_recall_w2	1.251973	0.280229	0.196764
rouge_f1_w2	1.231703	0.292081	0.203658
BLEURT	1.094424	0.37235	0.255333

Without preprocessing

	mse	pcorr	kendallcorr
bleu_w1	1.24208	0.286014	0.197938
bleu_w2	1.244415	0.284648	0.202001
rouge_recall_w1	1.260702	0.275125	0.192262
rouge_precision_w1	1.222024	0.297741	0.20983
rouge_f1_w1	1.217651	0.300297	0.209243
rouge_precision_w2	1.244839	0.2844	0.20088
rouge_recall_w2	1.268136	0.270778	0.190927
rouge_f1_w2	1.248619	0.28219	0.197477
BLEURT	1.082339	0.379416	0.26429

- EN-FI:

With preprocessing

	mse	pcorr	kendallcorr
bleu_w1	1.060834	0.361047	0.235384
bleu_w2	1.09294	0.340938	0.221034
rouge_recall_w1	1.092437	0.341252	0.221792
rouge_precision_w1	1.037889	0.375418	0.244203
rouge_f1_w1	1.051054	0.367172	0.238067
rouge_precision_w2	1.097963	0.337791	0.218634
rouge_recall_w2	1.123866	0.321567	0.208142
rouge_f1_w2	1.104184	0.333895	0.214382
BLEURT	1.101162	0.335788	0.216704

Without preprocessing

	mse	pcorr	kendallcorr
bleu_w1	0.898179	0.512997	0.341524
bleu_w2	1.065216	0.418989	0.293123
rouge_recall_w1	0.92588	0.497407	0.329099
rouge_precision_w1	0.881087	0.522616	0.348401
rouge_f1_w1	0.888932	0.518201	0.341341
rouge_precision_w2	1.05588	0.424243	0.296875
rouge_recall_w2	1.081815	0.409647	0.288378
rouge_f1_w2	1.062589	0.420468	0.292348
BLEURT	0.870491	0.52858	0.335507

- EN-ZH:

With preprocessing

	mse	pcorr	kendallcorr
bleu_w1	1.083773	0.422146	0.287555
bleu_w2	1.113169	0.40635	0.29359
rouge_recall_w1	1.132408	0.396011	0.269372
rouge_precision_w1	1.068441	0.430385	0.292041
rouge_f1_w1	1.057836	0.436083	0.295501
rouge_precision_w2	1.110168	0.407962	0.293991
rouge_recall_w2	1.131274	0.396621	0.286235
rouge_f1_w2	1.110509	0.407779	0.292018
BLEURT	1.585074	0.152767	0.09646

Without preprocessing

	mse	pcorr	kendallcorr
bleu_w1	1.824916	0.023886	0.004946
bleu_w2	1.80016	0.037189	0.019773
rouge_recall_w1	1.817211	0.028027	0.017531
rouge_precision_w1	1.808076	0.032936	0.017664
rouge_f1_w1	1.811713	0.030981	0.017586
rouge_precision_w2	1.830035	0.021136	0.011278
rouge_recall_w2	1.834927	0.018507	0.011262
rouge_f1_w2	1.831116	0.020555	0.011268
BLEURT	1.58651	0.151996	0.09581

- RU-EN:

With preprocessing

	mse	pcorr	kendallcorr
bleu_w1	1.269816	0.242116	0.155715
bleu_w2	1.249752	0.255568	0.172834
rouge_recall_w1	1.258468	0.249724	0.166822
rouge_precision_w1	1.222902	0.273569	0.183206
rouge_f1_w1	1.219188	0.276059	0.186087
rouge_precision_w2	1.252753	0.253556	0.169355
rouge_recall_w2	1.269918	0.242047	0.162321

Without preprocessing

	mse	pcorr	kendallcorr
bleu_w1	1.228187	0.311382	0.206475
bleu_w2	1.232012	0.309744	0.217681
rouge_recall_w1	1.251355	0.298923	0.205131
rouge_precision_w1	1.190027	0.333666	0.229459
rouge_f1_w1	1.195571	0.330525	0.226635
rouge_precision_w2	1.233501	0.309038	0.216881
rouge_recall_w2	1.266566	0.290306	0.204571

	mse	pcorr	kendallcorr
rouge_f1_w2	1.255521	0.2517	0.167682
BLEURT	1.105221	0.352468	0.238524

	mse	pcorr	kendallcorr
rouge_f1_w2	1.241795	0.304339	0.212457
BLEURT	1.09099	0.389512	0.274669

ZH-EN:

With preprocessing

	mse	pcorr	kendallcorr
bleu_w1	1.231177	0.327749	0.198846
bleu_w2	1.25521	0.314527	0.200727
rouge_recall_w1	1.405641	0.231763	0.141445
rouge_precision_w1	1.163599	0.364929	0.231118
rouge_f1_w1	1.22252	0.332512	0.203336
rouge_precision_w2	1.256264	0.313947	0.201395
rouge_recall_w2	1.344956	0.265151	0.166039
rouge_f1_w2	1.286034	0.297568	0.186136
BLEURT	1.171141	0.36078	0.227843

Without preprocessing

	mse	pcorr	kendallcorr
bleu_w1	1.264329	0.287864	0.191866
bleu_w2	1.263003	0.2887	0.194604
rouge_recall_w1	1.272151	0.283482	0.188592
rouge_precision_w1	1.215057	0.31611	0.213564
rouge_f1_w1	1.217686	0.314608	0.209664
rouge_precision_w2	1.253879	0.293924	0.199806
rouge_recall_w2	1.278568	0.279814	0.189689
rouge_f1_w2	1.257644	0.291773	0.196572
BLEURT	1.121538	0.369476	0.249322

3. GridSearchCV result: Model hyperparameters

cs-en	ensemble.GradientBoostingRegressor(learning_rate= 0.03, max_depth= 4, n_estimators= 200, subsample=0.8)
de-en	ensemble.GradientBoostingRegressor(learning_rate= 0.01, max_depth= 4, n_estimators= 300, subsample=0.7)
en-fi	lgb.LGBMRegressor(boosting_type='gbdt', learning_rate= 0.01, max_depth= 4, n_estimators= 300, subsample= 0.7,metric= 'rmse')
en-zh	lgb.LGBMRegressor(boosting_type='gbdt', learning_rate= 0.01, max_depth= 3, n_estimators= 300, subsample= 0.9,metric= 'rmse')
ru-en	lgb.LGBMRegressor(boosting_type='gbdt', learning_rate= 0.01, max_depth= 3, n_estimators= 800, subsample= 0.9,metric= 'rmse')
zh-en	lgb.LGBMRegressor(boosting_type='gbdt', learning_rate= 0.05, max_depth= 3, n_estimators= 200, subsample= 0.9,metric= 'rmse')