Text Mining Group Project
Building a metric for quality assessment of translations

Franz Michael Frank, Henrique Vaz, Philipp Metzger
20200618, 20200586, 20201058

Link to our GitHub Repository:
https://github.com/FranzMichaelFrank/Text_Mining_Group_Project

1 Introduction

People increasingly rely on algorithms in their day-to-day lives. This also applies regarding problems with foreign languages. Whereas in the past, dictionaries were laboriously browsed to find the right terms, nowadays all it takes is a quick Google search to find a desired translation. This is of course very convenient and timesaving. However, it is quite uncertain how good such a translation is in fact. This is the starting point of this project. The goal of this work is to develop a metric that is able to estimate the quality of a translation as accurately as possible. As a reference point, a rating assigned to the translation by human experts is used. The aim is to achieve the highest possible correlation between the developed metric and the human evaluation.

2 Methodology

In this project, the authors explored three different classes of metrics: lexical metrics, embedding based metrics, more specifically a metric called Word Mover's distance and trainable metrics, more specifically a linear regression model which uses vectors of sentence embeddings as input.

2.1 Methodology: Lexical metrics

In terms of lexical metrics, this study focused on the BLEU as well as the ROUGE metric. In total, 12 different variations of these two metrics were examined: BLEU Star Score (a naïve and very simply version of the BLEU Score), Sentence BLEU Score, Corpus BLEU Score, ROUGE 1 Precision, ROUGE 1 Recall, ROUGE 1 F-Measure, ROUGE 2 Precision, ROUGE 2 Recall, ROUGE 2 F-Measure, ROUGE L Precision, ROUGE L Recall and ROUGE L F-Measure. All the mentioned lexical metric variations were applied to six different language pairs. Subsequently, it was analyzed which metric is most suitable for each language pair by calculating the Pearson as well as the Kendall correlation of the computed score and the available z-score.

Next, a simple linear regression was performed, using two different approaches: one with the respective five best performing metrics per language pair as explanatory variables and one with all 12 metrics as explanatory variables. For this purpose, the respective data sets were split in an 80:20 ratio between training and development data. The performance was evaluated using the Pearson and Kendall correlation coefficients between the predictions for the development set and the corresponding actual z-scores.

Finally, regardless of whether it was a single metric or a regression model combining multiple metrics, for each language pair, the best method was selected in terms of correlation, taking both the Pearson as well as the Kendall's τ correlation coefficients into account. A function was implemented that receives a dataset and then first determines the language pair of the dataset and subsequently, based on that language pair, applies the best metric to calculate the score. The function returns the scores as a vector.

2.2 Methodology: Word Mover's distance

For the Embeddings based metrics, Word Mover's distance was chosen to be implemented by the authors. The library "Word Mover's distance" used for this purpose can be inspected at [1]. This library is an adaption of subsets of the "genism" library [2]. For applying Word Mover's distance, a dictionary that contains the words of the vocabulary as keys and their embedding vectors as values is needed. For this, first, the "glove" embeddings that were also used in the

practical classes of the Text Mining module at Nova IMS were tested. This yielded good results for all datasets, except for the one with the filename "en_zh.csv"[1], which is why the authors tried out other embeddings in order to solve this problem. The solution that was found to be working well was to use an adapted version of the implementation provided in the Jupyter notebook "LaBSE.ipynb" for creating embedding vectors for the whole vocabulary contained in all datasets' columns "reference" and "translation" and then to store these in a dictionary and with this data to create a Word Mover's model and to apply it to our data. This yielded good results for all datasets, wich will be presented in chapter 3.1.2.

For training the model, a train/development split with a ratio 80:20 ratio was performed and the model, which contains one trainable parameter was trained using the training set.

The trainable parameter has the following purpose: The library used for computing the Word Mover's distance between two sentences returns a distance of inf, if the distance cannot be computed, which can for example be the case when one of the sentences does not contain a word that exists in the vocabulary at hand. The way the authors decided to treat these inf values is to first replace them by -1 and then to compute the maximum max of all computed distances to then assign the value of $value = max * factor$ to the values that were inf originally. The value of $factor$ is the one parameter that the authors optimized for by computing the Word Mover's distances for all training data, and then computing the correlations (Pearson and Kendall) between the predictions and the column "z-score" for a range of values of $factor$. The range chosen was [-2, 2] with a granularity of 0.01. The results of this optimization are presented in chapter 3.1.2. Having found the optimal value for $factor$ based on the training data, the resulting model was applied to the development set in order to check if the good performance still held, which it did. These results will also be presented in chapter 3.1.2.

2.3 Methodology: Linear Regression

The goal of this approach is to create a model that is able to predict z-scores given sentence embedding vectors as inputs. These embeddings are achieved using LASER (Language Agnostic Sentence Representation). LASER is a method that in a way disregards the language of a sentence, focusing more on semantic equivalences (or differences) to represent them. The output of such method is a vector representing a sentence.

The main idea to train the model was to have a data frame with a schema looking like [Source | Reference | Translation | z-score]. That was achieved by loading the embedding arrays provided by the Professor Ricardo Rei. Additionally, the authors also decided to test if adding linear combinations of the original columns would increase the model's performance. The additional columns are the element-wise differences between source and reference, source and translation, and reference and translation. This way the model had more information in order to possibly predict more accurately.

The columns of resulting matrix for training the linear regression are as follows: Source, Reference, Translation, Source-Reference, Source-Translation, Reference-Translation. The resulting matrix shape is nxk with n being the number of sentences in the dataset and k = 6x1024 + 1, whilst the last column represents the column with the target z-scores.

Having this preprocessing terminated the authors could finally train the model. A train/development split with an 80:20 ratio was performed, and the model was trained. After having the model trained, the author did predictions on the development dataset and kept the result to further analyze and to compute correlations. The results of this approach are described in chapter 3.1.3.

3 Results and Discussion

3.1 Results

3.1.1 Results: Lexical metrics

The best performing respective metric resulting from the approach concerning lexical metrics described in chapter 2.1 are: Regression with all 12 metrics included for ru_en, cs_en and zh_en;

[1] These results are displayed in [A1] in the Appendix of this report.

ROUGE L F-Measure for de_en; Corpus BLEU Score for en_zh; ROUGE 1 Precision for en_fi. The corresponding correlations are presented in Figure 1.

	cs_en	de_en	en_fi	en_zh	ru_en	zh_en	total
Pearson	0.478206	0.344518	0.574585	0.414971	0.343241	0.377669	0.393903
Kendall	0.324866	0.242234	0.384611	0.290762	0.237125	0.250284	0.269016

Figure 1: Pearson correlation and Kendall's τ coefficients for the individual language pair datasets for lexical metrics

3.1.2 Results: Word Mover's distance

The training described in Chapter 2.2 yielded the Pearson correlation and Kendall's τ coefficients depicted in [A2] in the Appendix of this report. The value for $factor$ that was chosen is the arithmetic mean of the two depicted optima, which resulted in a value of $factor = 0.4150$.

Applying this model to the development set which, just like the training set combines data from all the language pair datasets with a ration of 80:20 for each pair, yielded a Pearson correlation coefficient of -0.3888 and a Kendall's τ coefficient of -0.2550.

To verify that the model not only performs well on average but also on the individual language pair partitions of the development dataset, it was applied to those individually, which yielded the results showed in Figure 2. Please note that the presented numbers are correlations between the "z-scores" that increase with the quality of a translation and a distance that decreases when the quality of a translation increases. Hence the negative correlation values.

	cs_en	de_en	en_fi	en_zh	ru_en	zh_en	total
Pearson	-0.451805	-0.358197	-0.626033	-0.429695	-0.347714	-0.360621	-0.388761
Kendall	-0.298608	-0.243782	-0.415418	-0.282416	-0.235153	-0.232396	-0.255047

Figure 2: Pearson correlation and Kendall's τ coefficients for the individual language pair datasets for Word Mover's distance

3.1.3 Results: Linear Regression

This chapter describes the results obtained by the model described in chapter 2.3. As presented in Figure 3, this implementation reached an overall Pearson correlation coefficient of 0.3433 and an overall Kendall's τ coefficient of 0.2284.

Assessing each language pair individually it can be concluded that there is some dependence on the model's performance on the language that is being processed.

	cs-en	de-en	en-fi	en-zh	ru-en	zh-en	total
Pearson	0.407624	0.268159	0.487035	0.360225	0.301637	0.300040	0.343252
Kendall	0.270616	0.184097	0.327127	0.231100	0.203054	0.201126	0.228372

Figure 3: Pearson correlation and Kendall's τ coefficients for the individual language pair datasets for linear regression

4 Discussion

Altogether, it is quite clear that both the experiment with the lexical metrics as well as the experiment with the Word Mover's Distance outperformed the linear regression approach described in chapter 2.3 and 3.1.3. Between the model using lexical metrics and the one using the Word Mover's distance, the correlations for the development set are very similar, whilst the first one scores slightly higher in magnitude for both the Pearson correlation coefficient and Kendall's τ coefficient (see Figure 1 and Figure 2).

Taking this and the fact that the Word Mover's model is one sound model that works well on all language pairs, whereas the lexical model contains multiple metrics that are being activated depending on the input language in a "hard-coded" way, the authors decided to make the Word Mover's distance model the final model of this project and to apply it to the test dataset provided by Professor Ricardo Rei.

5 Conclusion

As the main result of this project, it can be stated that there is a variety of different approaches to solve the task of assessing the quality of a translation. The authors found that the Word Mover's distance, along with a few adaptions, works well for all languages that have been tested and made the model using the Word Mover's distance their final translation quality assessment model for this project.

6 References

[1] https://pypi.org/project/word-mover-distance/#description, viewed on 27 May 2021, 16:34

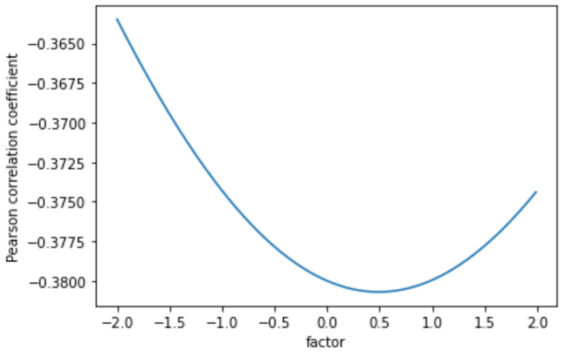
[2] https://github.com/RaRe-Technologies/gensim, viewed on 27 May 2021, 17:11

A Appendices

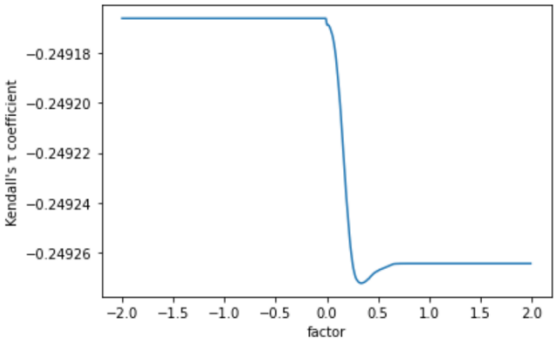
[A1]

Correlations that resulted in the authors' decision to look for different word embeddings (See chapter 2.2, the problematic correlation coefficients close to zero are written in bold):
cs_en: Pearson: -0.4296, Kendall: -0.2943
de_en: Pearson: -0.3089, Kendall: -0.2168
en_fi: Pearson: -0.3277, Kendall: -0.2288
en_zh: Pearson: **-0.0049**, Kendall: **0.0057**
ru_en: Pearson: -0.3065, Kendall: -0.2137
zh_en: Pearson: -0.2972, Kendall: -0.1957

[A2]



Pearson correlation coefficient for different values of $factor$



Kendall's τ coefficient for different values of $factor$