Machine Translation Metrics Shared Task

Federico De Alba (m20201015), John Ruiz (m20200540), Rupesh Baradi (m20200994)

1. Abstract

Rating a translation of a phrase is one of the most important tasks in machine translation since this is the way of knowing if improvements are being made and/or to compare different methods of translation.

There are several metrics to compare phrase similarity, but in this report, we exploited the combination of several different available of these metrics and some metrics created by us and then, creating a model combining all these metrics into just one score so we can get the highest possible correlation possible with the human judgements.

2. Introduction

Machine translation is a challenging task that traditionally involves large statistical models developed using highly sophisticated linguistic knowledge.

Judging how good or bad a machine translation is, is a hard thing to do and the best way to do it is to have an assessment from humans, due to the human capacity of abstraction of the sentences we are reading and not just the comparison of lexical-level features.

3. Creation of the Model

The first steps we took towards the task of creating a metric that predicts the quality of a machine translation was to first check how well the distance metrics and the orthography metrics correlated to the human judgement score.

Before calculating the metrics, we performed the standard pre-processing on the text, which includes the lower casing of all letters, removing punctuation, removing tokens with numbers on them, removing stop-words (with the help of nltk's list of stop-words for English and Finnish but no removal of stop words in Mandarin) and at the end, application of nltk's word lemmatizer.

An extra step we had to introduce in the Mandarin translation was the use of the JIEBA library, which helps in the tokenizing of texts written in Mandarin. This task cannot be done just by splitting the text on the blank spaces because in the Mandarin writing system not all words are divided by spaces.

Likewise, another step taken during preprocessing was introducing pymystem a morphological analyzer for Russian language improving the efficiency for lemmatization and punctuation remotion

After the pre-processing of the text, we then proceeded to calculate the metrics distance and orthography metrics. These metrics were:

3.1 Creating the Baseline

- Distance metrics:
  - Cosine similarity
- Orthography metrics:
  - Jaccard similarity
  - Dice similarity
  - Minimum Edit Distance

The results were not satisfying enough; by creating an XGBoost model to predict the "z-score" based on the metrics mentioned above, we were able to get the following results on a test set data made of 30% of the entire dataset:

Translation	Pearson Correlation	Kendall-Tau Correlation
Into English	0.352	0.240
Into Finnish	0.570	0.361
Into Mandarin	0.455	0.306

Table 1 – Results with first set of metrics

Correlation of all metrics are shown in Appendix A.

3.2 Creating the Models

The next step we took were to investigate what other metrics existed for sentence/phrase comparison and that were easily available for Python. The metrics that we then proceeded to calculate were the following:

- BLEU (Bilingual Evaluation Understudy): This metric works by counting matching n-grams in the candidate translation to n-grams in the reference text. We calculate the 1,2,3 and 4 cumulative n-grams BLEU scores.
- GLEU (Google-BLEU): Modified BLEU score; instead of averaging the sentence level GLEU scores (i.e. macro-average precision), sum up the matching tokens and the max of hypothesis and reference tokens for each sentence, then compute using the aggregate values.
- Ratio from Difflib: Ratio from 0 to 1. Calculated using the total number of elements in both sequences (T), and the number of matches (M), this is 2.0*M / T.
- Fuzzywuzzy: 4 different metrics developed by SeatGeek, which uses Levenshtein Distance in different ways (substring matching, ordering alphabetically and only using the complement between the 2 phrases) to calculate the distances between 2 different phrases.
- chrF: F-score based on character n-grams
- Meteor: Meteor evaluates a translation by computing a score based on explicit word-to-word matches between the translation and a given reference translation.

These metrics showed an overall good performance, improving the correlation score that we obtained with just the distance and orthography metrics used in the first section of this report.

Translation	Pearson Correlation	Kendall-Tau Correlation
Into English	0.371	0.252
Into Finnish	0.615	0.395
Into Mandarin	0.509	0.355

Table 2 – Results with second set of metrics

Correlation of each metric previously described are shown in Appendix A.

As we can see in Table 2, we had a slight increase in correlation, but still, not the best results that we knew we could get.

3.3 Word Embeddings

So, for the last step, we investigated word embeddings, representing the third model apart from the baseline relying on the distances metrics and the second on XGBoost. Training our own model for creating our own word embeddings would have taken a long time, so we decided to use the 100 dimensions Glove pre-trained word embeddings (just for English).

After having the word embeddings matrix, we created 5 new metrics for comparing sentences/phrases:

- Mean Euclidean distance between each word of two sentences/phrases.
- Median Euclidean distance between each word of two sentences/phrases.
- Mean cosine similarity between each word of two sentences/phrases.
- Area under the curve per number of words: For this metric, first we need to delete the common words between the 2 phrases to be evaluated. After, we calculate the cosine similarity using the word embeddings (100 dimensions) between each word and then sort the cosine similarities ascendingly. Example results are shown below:

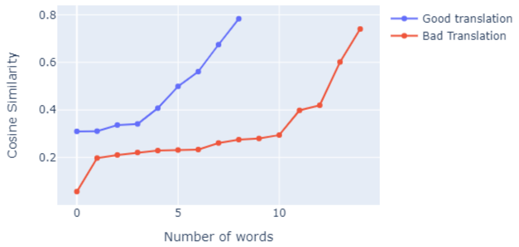

Fig 1 - Example of a good and bad translation

As we can imply, a good translation will have less different words and a higher cosine similarity, contrary to a bad translation, which will have more different words and a lower cosine similarity. To quantify this, we take the area under the curve and then divide it by the number of words to normalize it.

- Slope of the different words curve cosine similarity: Back to Figure 1, we can see that the bad translation's curve will have a higher slope than the good translation's curve. We will also use the calculated slope as one of our metrics.

Using these 4 new metrics, the correlation between the score predicted by our model and the one given by the annotators improves. Results are shown in Table 3.

Translation	Pearson Correlation	Kendall-Tau Correlation
Into English	0.391	0.265
Into Finnish	NA	NA
Into Mandarin	NA	NA

Table 3 – Results with third set of metrics

Correlation of each metric previously described are shown in Appendix A.

Finally, we looked into COMET, a neural network framework for training multilingual machine translation evaluation models, read the research paper described on the references and decided to mimic somehow in small scale the steps taken to predict on machine translations starting by finding a large-scale pretrained multilingual encoder and found that LaBSE was highly usable for this purpose, performing the following actions:

- Get the sentence embeddings for each of the translations.
- Concatenate the sentence embeddings.
- Apply feature engineering performing the element-wise product and absolute element-wise difference between the translations.
- The set of new features were concatenated and pushed into a Multi-Layer Perceptron Regressor (MLPR) neural network, two models were trained in parallel one with (3846 features), the second not using it (3074 features), the following hyper-parameters were used on the training process.

Hyper-parameter	Value (6-7 features)
Optimizer	Adam
Learning Rate	3e-05
Batch Size	16
Loss function	MSE
Feed-Forward Activation	Tanh
Feed-Forward Hidden Units	3072, 1536, 768

Table 4 – Model Architecture

The training data was split into 60% training, 20% validation and 20% testing, the correlation between the score predicted by our first model based on the 6 main features and the one given by the annotators improves. Results are shown in Table 4

Translation	Pearson Correlation	Kendall-Tau Correlation
CZ-EN	0.527	0.371
DE-EN	0.392	0.271
RU-EN	0.401	0.268
ZH-EN	0.415	0.292
EN-FI	0.616	0.407
EN-ZH	0.525	0.363

Table 5 – Results of LaBSE

The correlation between the score predicted by our first model on 6 features and the one given by the annotators improves significantly, but we decided not to choose this LaBSE because of the long time it takes for training (several hours) and we were not able to experiment more because of this.

We believe the result we obtained with using the combination of several different similarity metrics is a good tool and can compete closely to the LaBSE results, as we can see in the results above.

4. Conclusion

In this report, we showed how we were able to predict the score given by a human annotator to a translation with a decent degree of correlation. By using several different approaches to the machine translation judgement task, we increased the correlation in comparison of using these metrics by themselves. One metric can be very good at catching certain kind of things that other metric may be very bad at it, and that is why combining several different metrics was the way we chose to approach this problem.

5. References

[1] Python's difflib Library, https://docs.python.org/3/library/difflib.html

[2] Bird, Steven, Edward Loper and Ewan Klein (2009), *Natural Language Processing with Python*. O'Reilly Media Inc.

[3] Cohen, Adam (2011). FuzzyWuzzy: Fuzzy String Matching in Python, https://chairnerd.seatgeek.com/fuzzywuzzy-fuzzy-string-matching-in-python/

[4] Popovic, Maja (2015). chrf: character n-gram F-score for automatic MT evaluation, http://www.statmt.org/wmt15/pdf/WMT49.pdf

[5] Lavie, Alon and Agarwal, Abhaya (2004). METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments, http://www.cs.cmu.edu/~alavie/METEOR/pdf/Lavie-Agarwal-2007-METEOR.pdf

Appendix

A. Individual Pearson correlation of each metric used to make the final models.

Metric	de_en	cs_en	ru_en	zh_en	en_fi	en_zh
cosine_similarity	0.289	0.377	0.304	0.288	0.505	0.393
jaccard_similarity	0.290	0.380	0.296	0.302	0.477	0.361
dice_similarity	0.273	0.359	0.287	0.237	0.491	0.371
bleu1	0.299	0.390	0.314	0.313	0.514	0.399
bleu2	0.270	0.349	0.294	0.279	0.413	0.383
bleu3	0.244	0.288	0.248	0.236	0.309	0.323
bleu4	0.210	0.234	0.208	0.205	0.214	0.261
difflib_ratio	0.293	0.395	0.301	0.301	0.486	0.448
fz_r	0.293	0.395	0.301	0.301	0.486	0.448
fz_pr	0.281	0.368	0.288	0.295	0.510	0.428
fz_tsor	0.280	0.368	0.282	0.284	0.497	0.421
fz_tser	0.280	0.368	0.292	0.277	0.565	0.400
MED	-0.160	-0.174	-0.181	-0.202	-0.338	-0.221
chrf	0.289	0.368	0.298	0.273	0.559	0.380
gleu	0.285	0.371	0.289	0.293	0.459	0.370
meteor	0.289	0.362	0.300	0.287	0.468	0.384
mean_distance_using_embedding	-0.090	-0.116	-0.104	-0.162	NA	NA
median_distance_using_embedding	-0.055	-0.070	-0.067	-0.108	NA	NA
cossim_mean	0.074	0.018	0.112	0.125	NA	NA
cossim_area	-0.032	-0.098	-0.003	0.036	NA	NA
cossim_slope	-0.071	-0.057	-0.012	-0.155	NA	NA