Multilingual Translation Quality Estimation

Unfollow 2 Edit

Jéssica Silva Feb 17

- **Goal**: Assess the quality of a translation system/human without access to reference translations.
- **Output**: Translation Quality score between 0 and 1 at sentence-level
- **Setup**: Multilingual

Applications:

- Reduce **post-editing effort**: Post-editing means that a human translator is editing or revising text that has been translated by machine translation software
- Filter **bad translations**: filter bad translations of a translation batch

Feature:

- 34059 **[NLP][NLPMetrics] Machine Translation Quality Estimation Score** Implementation stage

Contents

1. Multilingual Sentence Embeddings

```
Sentence Embeddings: are the representation of the meaning of a sentence in low-dimensional vectors, which capture both the syntax and semantics of the text corpus.
```

The **Multilingual Sentence Embeddings** is a resource that can be used to measure the quality of a translation, through the semantic similarity between sentences in different languages. As embeddings are dense numerical vectors, it is possible to capture the **semantic similarity** through the cosine distance/similarity between two vectors.

We chose to generate sentence embeddings through a state-of-the-art architecture called **Sentence Transformer**, launched in 2020 at the EMNLP (Empirical Methods in Natural Language Processing) conference.

The translation quality can be measured at the sentence level:

- **Sentence-level**: The goal of the Sentence-level Quality Estimation (QE) task using multilingual sentence embeddings is to predict the quality of the whole translated sentence using the cosine similarity score between two sentences in different languages.

2. Architecture

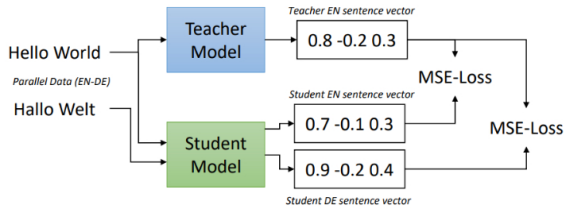
2.1. Sentence Transformer (Distillation approach)

```
Knowledge distillation: describes the process to transfer knowledge from a teacher model to a student model. It can be used to extend sentence embeddings to new languages.
```

The Sentence Transformer (Paper) architecture consists of two models, the **Teacher** and the **Student**:

- **Teacher model**: produces sentence embeddings from source language.
- **Student model**: Using translated sentences, this model needs to mimic the teacher and generate sentence embeddings in the target language.

This training will produce an alignment of vector spaces (**source** and **target**) making it possible to measure the cosine distance between them.



The architecture used here is a **Siamese network**. A siamese network is a neural network trained to compare the similarity between two inputs. The key is that it generates internal representations of the sentences, suited for similarity problems.

Some features:

- Allows to create multilingual versions from previously monolingual models
- An easy and efficient method to extend existing sentence embedding models to new languages
- The training is based on the idea that a translated sentence should be mapped to the same location in the vector space as the original sentence

2.2. Language Models (Transfer Learning approach)

```
Transfer Learning: an approach where a model developed for a task is reused as the starting point for a model on a second task.
```

Sentence Transformers models works with transfer learning and needs to be initialize with some pretrained deep neural language models as **BERT**, **GPT 2 or 3**, **XLM**, **XLNet**, **RoBERTa** and so on.

We follow the Sentence Transformer Paper and kept:

- **SBERT** (Paper) initializing the **teacher model** (English model)
- **XLM-R** (Paper) initializing the **student model** (Multilingual model with 100 languages)

2.3. Quality Estimation Models

2.3.1. Baseline model

We used the Sentence Transformers python library provided by the UKPLab with the following pre-trained model:

- **distiluse-base-multilingual-cased-v2**: Multilingual knowledge distilled version of multilingual Universal Sentence Encoder.

2.3.2 Custom model (Fine-tuning approach)

```
Fine-tuning: Each task is unique, and having sentence embeddings tuned for that specific task greatly improves the performance.
```

Addressing as a Binary Classification Problem

As DC Translation validation data (Sect 3.3) has binary labels, **good** (True) and **bad** (False) **translations**, the fine-tuning of the model was done through a **binary classifier**. For this, we added a new layer on the top of the architecture with the loss function chosen for this purpose.

Loss Function

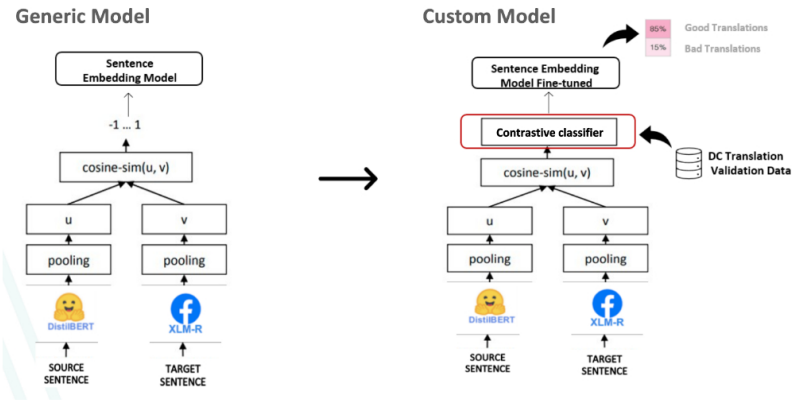
The goal is fine-tune our sentence transformers model to generate meaningful sentence embeddings using the Translation validation data. The loss function determines how well our embedding model will work for the specific downstream task. To fine-tune in a binary task, we can use:

- SoftmaxLoss
- **ContrastiveLoss**
- OnlineContrastiveLoss

We tested all available Loss options and got better results with **Contrastive loss**.

Contrastive Loss: Expects as input two texts and a label of either 0 or 1. If the label == 1, then the distance between the two embeddings is reduced. If the label == 0, then the distance between the embeddings is increased.

The following picture shows the neural architecture of those models.



- **Baseline model**: The Generic model trained on public translation dataset (Sect 3.2) (distiluse-base-multilingual-cased-v2)
- **Custom model**: The Generic model fine-tuned on DefinedCrowd translations (Sect 3.3) (DC Translation Validation Jobs)

2.4. Quality Estimation Metric (Cosine Similarity)

```
Cosine similarity: measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors (range: [0,1]).
```

As one of the requirements of the project was the development of a metric that deals with the quality of **human translations**, we chose to work with **cosine similarity** between sentence embeddings, to find sentences with a similar meaning in different languages.

This is a common approach in the **Multilingual Quality Estimation task**, when trained in a **self-supervised** way.

3. Data

3.1. Language Coverage

These Quality Estimation models supports 50+ languages and more languages can be added by model extension.

Supported languages: ar, bg, ca, cs, da, de, el, es, et, fa, fi, fr, fr-ca, gl, gu, he, hi, hr, hu, hy, id, it, ja, ka, ko, ku, lt, lv, mk, mn, mr, ms, my, nb, nl, pl, pt-pt, pt-br, ro, ru, sk, sl, sq, sr, sv, th, tr, uk, ur, vi, zh-cn, zh-tw.

3.2. Public Training Data

OPUS website (http://opus.nlpl.eu/)

Translation Datasets

Dataset	Description
GlobalVoices	A parallel corpus of news stories from the web site Global Voices.
TED2020	Translated subtitles crawled for about 4,000 TED talks, available in over 100 languages. This dataset is available in our repository.
NewsCommentary	Political and economic commentary crawled from the web site Project Syndicate, provided by WMT.
WikiMatrix	Mined parallel sentences from Wikipedia in different languages (Schwenk et al., 2019). We only used pairs with scores above 1.05, as pairs below this threshold were often of bad quality.
Tatoeba	Tatoeba is a large database of example sentences and translations to support language learning.
Europarl	Parallel sentences extracted from the European Parliament website (Koehn, 2005).
JW300	Mined, parallel sentences from the magazines Awake! and Watchtower (Agic and ´Vulic´, 2019).
OpenSubtitles2018	Translated movie subtitles from opensubtitles.org (Lison and Tiedemann, 2016).
UNPC	Manually translated United Nations documents from 1994 - 2014 (Ziemski et al., 2016).

Dictionaries to enrich low-resource language pairs

Dictionary	Description
MUSE	MUSE provides 110 large-scale ground-truth bilingual dictionaries created by an internal translation tool (Conneau et al., 2017b).
Wikititles	Wikipedia database dumps to extract the article titles from crosslanguage links between Wikipedia articles. For example, the page "United States" links to the German page "Vereinigte Staaten". This gives a dictionary covering a wide range of topics.

Vocabulary per language

- SBERT (English): vocabulary size of 30k mainly consisting of English tokens
- XLM-R (100 languages): Vocabulary with 250k entries from 100 different languages

3.3. DC Training Data

Translation Validation is on of the steps for the Translation Process adopted by DefineCrowd.

In this step, a randomly sample is taken from the translation batch (usually, around 1,000 sentences) and configured in the platform to be validated by crowd members. To perform this validation, each crowd members is required to answer three questions:

1. source-intelligible: **Is the source intelligible?** (**True** or **False**)
2. accuracy: **Is the meaning of the source conveyed?** (**True** or **False**)
3. fluency: **Are translations fluent and sound natural in the target language?** (**True** or **False**)

The data used in this research is a collection of manually picked translation validation jobs from the Crowd Platform. The following jobs are part of this data:

Translation Validation Jobs

Job id	Language Pair	Hits	Total	Intelligible	Accuracy (%)	Fluency (%)
6195	English-Italian	2276	2874	2859	96.3	95.1
5478	English-Italian	1000	1000	971	94.4	92.7
18954	English-Italian	119	357	354	98.6	96
6194	English-Russian	920	986	947	88.2	69.1
5510	English-Russian	1000	3000	2937	92.7	82.8
8262	English-Russian	1000	3000	2983	93.6	81.2
18957	English-Russian	119	357	351	95.4	86.9
19705	English-Russian	1000	3000	2932	88.9	43.8
5477	English-Arabic	1000	1000	966	95.9	84.6
6198	English-Arabic	2289	2864	2752	90.0	74.6
5479	English-Japanese	1000	1000	908	72.6	39.4
5526	English-Japanese	1000	2000	1934	98.6	93.1
6197	English-Japanese	2404	3010	3004	99.8	97.5
19704	English-French	1000	3000	2940	92.2	88.7
6861	French-English	1000	3000	2990	96.1	89.3
6743	Korean-Chinese	1000	3000	2994	95.7	91.2
6763	Japanese-English	1000	2011	1992	91.2	88.8
20609	Japanese-English	999	2997	2982	89.7	65.8
6778	Korean-Japanese	1000	3000	2984	97.3	91.2
6708	English-German	1000	3000	2982	92.9	89.3
6866	English-German	1000	3000	2948	90.7	78.3
18950	English-German	119	357	357	98.6	95.2
19702	English-German	1000	3000	2951	95.7	77.7
6859	English-German	1000	3000	2876	94.2	84.9
6741	Japanese-Chinese	1000	3000	2999	95.4	93.3
20560	Japanese-Chinese	1000	3000	2956	94.4	89.4
6740	Japanese-Korean	1000	3000	2999	97.9	94.9

18952	English-Japanese	119	357	355	93.2	82.8
19681	English-Japanese	1000	3000	3000	99.8	97.6
19684	English-Japanese	999	2997	2873	84.2	70.1
19723	English-Japanese	1000	2999	2549	78.3	55.7
19725	English-Japanese	1000	2999	2794	79.4	50.8
19726	English-Japanese	1000	2999	2879	74.3	47.6
19729	English-Japanese	1000	2999	2680	84.1	58.8
5499	English-Korean	1000	3000	2900	97.1	90.0
6196	English-Korean	2409	3002	2814	90.3	77.4
6744	English-Korean	1000	3000	2986	86.1	82.5
18953	English-Korean	119	357	326	94.5	91.7
6222	Korean-English	2944	3137	2941	88.5	79.8
6742	Korean-English	1000	3000	2982	94.3	89.3
5215	Portuguese-Chinese	1000	1000	959	84.5	70.2
5211	Russian-Chinese	1000	1000	1000	93.6	82.2
5214	Russian-Chinese	1000	1000	985	82.2	63.0
6863	Russian-English	1000	2728	2592	88.8	77.3
6709	English-French	1000	3000	2981	97.4	93.9
18951	English-French	119	357	351	97.2	94.9

18949	English-Spanish	119	357	343	93.0	77.3
19703	English-Spanish	1000	3000	2993	95.9	89.3
6860	Spanish-English	1001	3001	2638	85.0	77.0
8272	English-PortugueseBr	1000	3000	2994	96.9	92.2
19700	English-PortugueseBr	1000	3000	2994	99.0	97.4
6862	PortugueseBr-English	1000	2999	2661	89.8	76.6
18955	English-Chinese	119	357	346	94.2	90.2
19699	English-Chinese	999	2997	2788	92.6	75.1
6865	Chinese-English	1000	3000	2841	73.3	55.2
8434	Chinese-English	1000	3000	2979	81.4	62.4
8446	Chinese-English	1000	3000	2809	94.4	86.9
8261	Chinese-Japanese	1000	3000	2989	95.4	88.6
8388	Chinese-Japanese	1000	2999	2999	99.0	92.5
19662	Chinese-Japanese	1000	3000	2999	95.3	61.7
19674	Chinese-Japanese	1000	3000	2774	73.5	59.2
19675	Chinese-Japanese	1000	3000	2774	99.4	93.2
19676	Chinese-Japanese	1000	3000	2957	84.7	63.5
19682	Chinese-Japanese	1000	3000	2986	92.5	70.9
19686	Chinese-Japanese	1000	3000	2957	85.6	67.8

From the data collection, we generated a single CSV file with the all translation validation jobs (`data/processed/validation/dataset.csv`):

	language_pair	source	target	accuracy	jobid	agreement
59140	ru-en	Утверждается, что садист,ударивший девушку кулаком в живот, это сотрудник пятого батальона 2-го оперативного полка главного	It has been confirmed that the sadist who hit his girlfriend in the stomach with a fist is an employee of the fifth battalion of the 2nd operational regiment of the main	True	6863	True
25708	ru-zh	В репертуаре коллектива более 20 концертных программ, рассчитанных на разные зрительские и возрастные аудитории.	该团的曲目包括20多个适合不同年龄观众的音乐会节目。	True	5214	False
1033	en-ko	But I'm from Dallas so maybe I'm special	근데 내가 댈러스 출신이라 유독 그런 겁지도	True	6744	True
21936	jp-zh	本気で話したい方向けの英会話教材【JJ ENGLISHエクササイズ】	为真正想说英语的人准备的英语口语教材【JJ ENGLISH练习】。	True	20560	True
26250	ru-zh	Сервисный центр › 2-й Лесной переулок	服务处Lesnaya巷2号	True	5214	False
43065	es-en	Cuando eres el lambon del jefe // cuando te echan como un perro	When you suck up to the boss // when they kick you out like a dog	True	6860	True
17613	jp-en	１２月７日（土）、８日（日） 熊本県大会	December 7 (Sat.), 8 (Sun.) Kumamoto Prefectural Tournament	True	20609	True
29306	en-de	On November 18, 1944 we took off from France fifteen miles northwest of Paris to bomb	Am 18. November 1944 sind wir von Frankreich aus fünfzehn Meilen nordwestlich von Paris losgezogen, um zu bombardieren	True	19702	True
7653	zh-jp	昨天早晨就这么不欢而散，我也真的很痛苦。	昨日の朝、あんなに気まずくて別れてしまい、私も本当に辛いです。	True	8388	True
37382	zh-en	J V那么便宜？	Is J V so cheap?	True	6865	True

To test our models in a **balanced scenario**:

- held-out set consisting of some translations (source and target) in different languages (`data/processed/validation/test_balanced.csv`)
- balanced dataset in terms of the translation quality distribution (good and bad translations) and it was not used for model training (concept of 'unseen dataset')

To test our models in a **unbalanced scenario**:

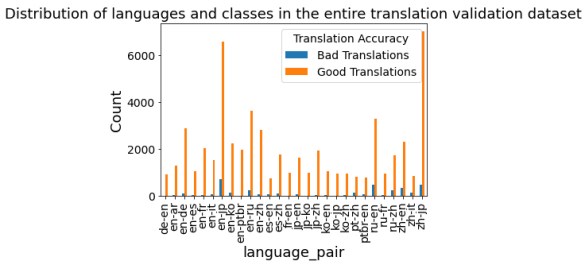
- test set consisting of some translation jobs in different languages (`data/processed/validation/test_unbalanced.csv`)
- unbalanced dataset in terms of the translation quality distribution (good and bad translations) and it was not used for model training (concept of 'unseen dataset')

To train our custom model:

- training sets without the balanced and unbalanced test sets
- balanced by class and language pair using a **undersamping strategy**
- split into train and dev sets
- **first custom models**: balanced scenario (without the first test set)
- **second custom models**: unbalanced scenario (without the second test set)

In summary, in this project, we have four different datasets:

- **entire set**: all the translation validation jobs (~60K sentence pairs).
- **training set**: dataset portion used to train our custom model. We balance this data using a undersamping strategy and split it into train and dev sets (~6K sentence pairs).
- **balanced test set**: dataset portion used to test our models. It only contains translations with rater's agreement and it is balanced using a undersamping strategy (~650 sentence pairs).
- **unbalanced test set**: translation jobs used to test our models. It is unbalanced data, following a real world distribution (~7K sentence pairs).



Regarding the distribution, as we can see in the figure below, we have a class imbalance problem for our metric of interest (translation accuracy):

3.4. Preprocessing steps

Tokenization tools

- SBERT (English): Wordpiece
- XLM-R (100 languages): SentencePiece (avoids language specific pre-processing)

4. Evaluation

To compare different models, we plot the **ROC curve** and measure the **area under the curve (AUC)**. We also measure the performance in a classification setup, using **Precision**, **Recall** and **F1-score** metrics.

4.1. Our models x Benchmark

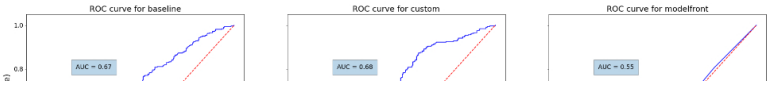
```
Benchmarking: process of measuring the performance of a company's product against our products.
```

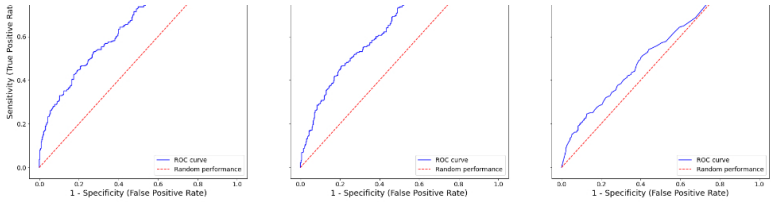
ModelFront is a technology company that offers scalable solutions for translation quality estimation. The ModelFront model gives us a translation risk prediction score and is based on deep learning techniques.

Comparison	Models
Our models	**baseline** and **custom models**
Benchmark	ModelFront model

Balanced Test set

ROC curve





Conclusions:

- By inspecting the behaviour for the ROC curve, we can see that the area under the curve (AUC) for our models ('baseline' and 'custom') are higher when compared with ModelFront. This demonstrate that our models have better performance on predicting the translation quality.
- ModelFront ROC curve is close to Random which shows a poor capacity for this model to fit the problem.
- The performance of the 'custom' model is very close to the performance of the 'baseline', indicating that we should provide more data for the training.
- All ROC curves are far from 1 (best score) which shows the complexity of the problem and the the difficulty in predicting the right labels.

Precision, **Recall** and **F1-score**

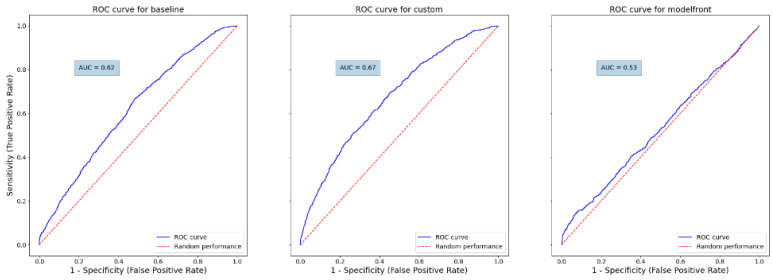
	model	accuracy	Bad Translation						Good Translation					
			# class	# correct	# incorrect	Precision	Recall	F1	# class	# correct	# incorrect	Precision	Recall	F1
0	baseline	0.63	322	148	67	0.69	0.46	0.55	322	255	174	0.59	0.79	0.68
1	custom	0.63	322	282	200	0.59	0.88	0.70	322	122	40	0.75	0.38	0.50
2	modelfront	0.56	322	79	41	0.66	0.25	0.36	322	281	243	0.54	0.87	0.66

Conclusions:

- We observed a better capability in our 'custom' model to balance between precision (0.59) and recall (0.88) which lead to a good avg f1-score (0.70) for the 'bad translation' class.
- The 'custom' model has a high Recall in the 'bad translation' class (threshold optimized for Recall)
- The 'custom' model has low Recall and high Precision in the 'good translation' class (Precision-Recall trade-off)
- We observe the ModelFront tendency to everestimate the translation quality. As it includes most of the examples as good translation, it fails more on predicting the bad translations (low recall - 0.25 and high error rate for good translations - # incorrect)

Unbalanced Test set

ROC curve



Conclusions:

- The 'custom' model has a better performance (0.67) compared to 'baseline' (0.62) and ModelFront (0.53).
- For all job batches, the performance of our models ('baseline' and 'custom') was superior to the performance of ModelFront.
- The performance of the 'custom' model is very close to the performance of the 'baseline', indicating that we should provide more data for the training.

Precision, **Recall** and **F1-score**

	model	accuracy	Bad Translation						Good Translation					
			# class	# correct	# incorrect	Precision	Recall	F1	# class	# correct	# incorrect	Precision	Recall	F1
0	baseline	0.59	743	503	2462	0.17	0.68	0.27	5925	3463	240	0.94	0.58	0.72
1	custom	0.68	743	385	1757	0.18	0.52	0.27	5925	4168	358	0.92	0.70	0.80
2	modelfront	0.61	743	349	2179	0.14	0.47	0.21	5925	3746	394	0.90	0.63	0.74

Conclusions:

- The 'custom' model has a high recall and precision for the majority class (good translation), resulting in the highest F1-score for that class.
- If we compare the metrics for the 'bad translation' (the class we used to optimize our threshold choice), with the exception of the 'en-de' pair, the 'baseline' has a higher recall score for this class (it gets more 'bad translations' - our optimization choice). For the 'en-de' language pair, ModelFront has a higher recall, but also the double of incorrect examples in this class (half of the precision).
- The 'baseline' model has better f1-score values for the 'bad translation' class when comparing all models.
- The 'custom' model has better f1-score values for the 'good translation' class when comparing all models.

4.1.1. Jupyter Notebook

Access the Jupyter Notebook used to create the results here.

4.1.2. Git Repository

Access the project git repository here.

5. Conclusions

- Better results for translation quality estimation using the DC data when compared with the commercial solution ModelFront
- Good Language Coverage (50 languages) with evaluation on 28 different language pairs (DC data)
- Allows training of a custom model, requiring few data for fine-tuning (~7,000 sentence pairs (DC data) splitted into train, dev and test sets)
- Works very well with human translations as it is based on the semantic similarity between sentences