Use transfer learning based model to conduct quality estimation for machine translated sentences

Mingming Zhang
Georgia Institute of Technology
Atlanta, GA USA
mzhang607@gatech.edu

He Yao
Georgia Institute of Technology
Atlanta, GA USA
hyao66@gatech.edu

Bofei Yu
Georgia Institute of Technology
Atlanta, GA USA
byu93@gatech.edu

Jingyao Zhu
Georgia Institute of Technology
Atlanta, GA USA
jzhu398@gatech.edu

Abstract

Translation service is important in today's life. A good Quality Estimation (QE) system is needed to assess the level of quality for certain translations provided to users. Here, we investigate an ensemble learner-based QE system from a WMT18 submitted work, which utilizes the predictions from several language models, such as NeUral Quality Estimation and Predictor-Estimator models. Furthermore, we conduct transfer learning with different types of strategies to the Predictor-Estimator-based models to see how the performance of the QE system might get affected. We hope the exploration can help reveal the potential improvements that can be brought to the state-of-art QE models. The performance of the baseline model and the transfer learned models will be evaluated against the same data sets.

1. Introduction/Background/Motivation

Translation services play a vital role in today's multilingual environment. With a lot of machine translation (MT) services openly available, a quality estimation (QE) system to objectively evaluate the quality of the translation is needed. In reality, the lack of good QE systems is the bottleneck to making MT services to be accountable and reliable. This project aims to explore the possible improvements that can be brought to build a quality estimation model to predict the quality of the MT sentences without any help from human intervention.

QE as a task of evaluation translation system's quality is a relatively new area to machine learning field. Some early work uses the QUETCH+ system, which combines a linear feature-based classifier with a feed-forward neural network, to predict labels of individual words [7]. Later, the *Unbabel* team improves the QUETCH+ system, by replacing a word-level linear classifier with a sentence-level first-order sequential model and extend three different neural network systems and two recurrent models with multiple instances of each model [1]. Later, researchers propose to add a two-stage model called Predictor-Estimator, which uses multi-level task learning for translation QE. The strength of using such model is combining a word prediction model and QE model [9, 10]. The Predictor-Estimator based model with other models in an ensemble fashion becomes one of the mainstream structures in QE research because it can leverage advantages of different models [11, 5].

We experiment with three transfer learning strategies. In the first strategy, we applied additional data sets to transfer learn our baseline algorithms. In the second strategy, it is around the limitation of the currently trained baseline model. We have tried words that are used less frequently and words that are used more frequently. For less frequent words, fewer sentence pairs are available in the data set. Due to the lack of exposure, the estimator suffers from a shortage of knowledge. Hence we selectively sample sentence pairs consisting of common words and applied transfer learning to this sampled data set. A better prediction of common words can improve the overall model performance as these words are more likely and frequently to show up in the evaluation task and real-life practice. As a result, a second transfer learning strategy, which relies on sentence pairs consisting of common words, is initiated and experiment with the study. In the last strategy, we select sentences that are less like each other by calculating Euclidean distance.

We will use the ***data*** and pre-trained ***models*** provided

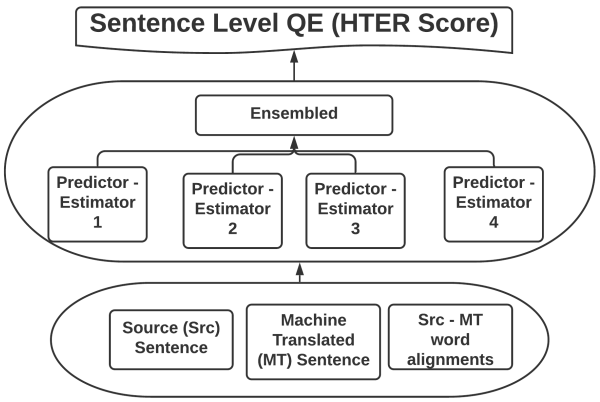

Figure 1. Sentence level QE, Our reproduced work flow and model structure based on submitted work by *Lucia, et al 2018* [11]. The model is an ensemble of several models. 'Predictor-Estimator 1' to 'Predictor-Estimator 4' are all Predictor-Estimator based models. 'P-E1' and 'P-E4' are the same structure but different instances; 'P-E2' and 'P-E3' are the same structure but different instances.

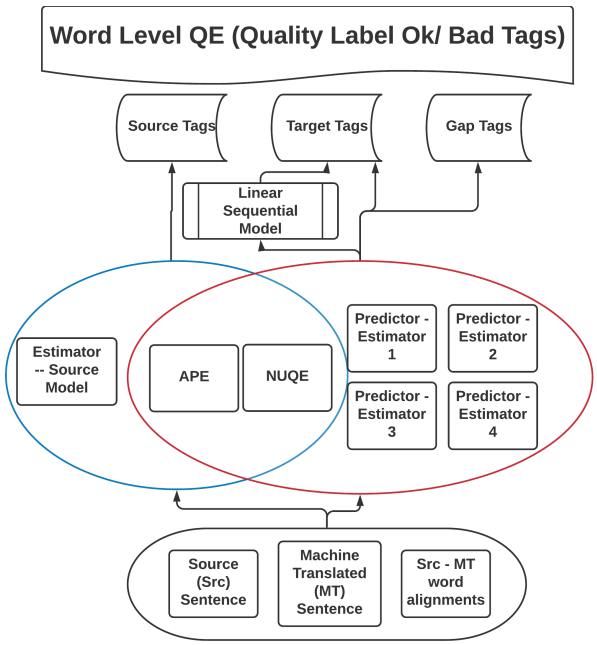

Figure 2. Word level QE, Our reproduced work flow and model structure based on submitted work by *Lucia, et al 2018* [11]. The model is an ensemble of several models. 'APE' is automatic Post-Editing adapted model for QE. 'NuQe' is NeUral Qaulity Estimation model. 'Predictor-Estimator 1' to 'Predictor-Estimator 4' are all Predictor-Estimator based models. 'P-E1' and 'P-E4' are the same structure but different instances; 'P-E2' and 'P-E3' are the same structure but different instances. And an Estimator model for source tag. The outputs of target tags from P-E 1-4 together with APE, NuQE are feed into Linear Sequential Model. The outputs are quality 'OK'/'BAD' labels for Source, Target and Gap tags. Please refer to Figure 3 for details.

in a WMT18 submission [11], because this work is with a mainstream Predictor-Estimator based ensemble models. Detailed instruction on how to set up the computer to reproduce the original work from this group can be found ***here***. All of the work is running on *OpenKiwi* with version 0.1.2, which is a version of an open-source framework built for quality estimation, developed by *Unbabel* team [4]. The data set contains the training and development data for the WMT18 QE task for the language translation from **English** to **German**. The data set contains sentence-level data and word-level data for a total of 13442 sentence pairs for training data and 1000 sentences for development data. For more details of the data set, please refer to Table 1. One thing we want to point out during the reproduction of the work is that we found the 755th, 776th, and 889th sentences have word alignment errors between the source-target sentence pairs. Thus, we have to remove these three sentence pairs and the final development data set used for evaluation in this project contains 997 sentence pairs.

For transfer learning the models, we add new data sets of English-German sentence pairs from ***WMT17***, ***WMT19*** and ***WMT20*** shared tasks. Using WMT17, 19, and 20 data sets, we come up with three different strategies. Corresponding to these three strategies, there are three different data sets. One of the data sets is that the English-German sentence pairs data set from WMT19 and WMT20 are combined, with a total of 20442 sentence pairs in training data and 2000 sentence pairs in the development data set. From the above big data set, we come up with a new subset named "common-word training and development". 11375 sentence pairs in training data and 988 sentence pairs in development data of common-word data set. In addition, there is another data set called "challenge training and development". Respectively, this training set has 2500 sentences in WMT17, WMT19, and WMT20 and the sentence pair is least likely based on applying Euclidean distance. The development set of this data set includes WMT17, WMT19, and WMT20 with 3000 sentence pairs. Further data selection used for transfer learning will be discussed in sections ***Approach*** and ***Experiments and Results***. To reproduce the work mentioned in this paper, please refer to the instruction mentioned ***here*** in our *GitHub* site.

2. Approach

Instead of training a QE model from scratch, we will leverage the state-of-art work that has been done in the QE systems for MT languages by using transfer learning techniques to investigate the possible improvements that can be

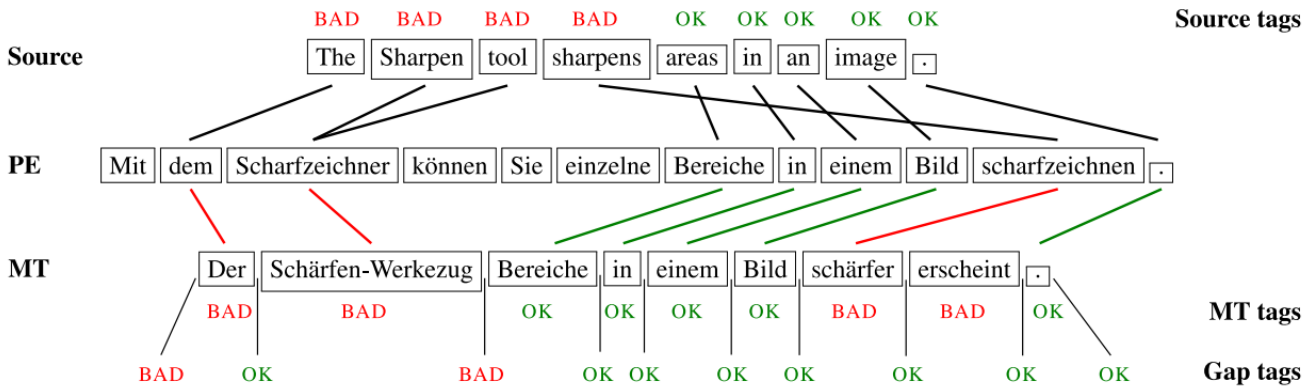

Figure 3. Tags for source, PE and MT sentences. Image adapted from *Fábio Kepler, et al 2018.* The source and MT tags are a sequence of 'OK' or 'BAD' labels. These tags are compared to PE sentences modified by a human expert to indicate whether a word in source or MT sentences are translated with problem [4].

Data file name (sentence level)	**Description (sentence level)**	**Data file (word level)**	**Description (word level)**
train/dev.nmt.src	source sentence	train/dev.src	source sentence
train/dev.nmt.mt	machine translated sentence	train/dev.mt	machine translated sentence
train/dev.nmt.hter	hter between translated and post-edited sentences, capped between 0 and 1	train or dev.tags	OK/BAD tags for MT tokens and gap tokens (one at the start of sentence and one after every token)
train/dev.nmt.pe	post-edited machine translation	train/dev.pe	post-edited machine translation
train/dev.nmt.ref	reference translation (not pre-processed)	dev.nmt.ref	reference translation (not pre-processed)
train/dev.nmt.add-labels	additional labels collected during post-editing	train/dev.src-tags	OK/BAD tags for source tokens that are aligned to target tokens
-	-	train/dev.src-mt.alignments	alignments between source and MT tokens (source-MT)

Table 1. Attributes contained in the training and development data sets.

brought to the QE models. This is the reason why we choose the submitted work from WMT18, as it is one version of ensemble models based structure with Predictor-Estimator models [11].

In order to have a good understanding of the workflow and model structure, We spend a decent amount of effort to reproduce the work by rewriting the code script for the prediction and evaluation part of the WMT18 submitted work into an Ipython notebook, while the original work is using shell file to run the evaluation. Fig.1 and 2 show our reproduced workflow and the model structure. Figure 1 shows how Predictor-Estimator models can take source and MT sentences to generate sentence-level QE. Figure 2 shows our reproduced ensemble-based model and it has automatic Post-Editing (PE) adapted model 'APE', NeUral Quality Estimation model 'NuQe' and 4 Predictor-Estimator based models. The 'APE' model is trained on human post-edits [2], and in this project its outputs are used as pseudo-post edits to generate word-level quality labels for source PE sentences and MT PE sentences. 'NuQe' jointly learns the MT tags and source tags, and here it predicts the probability or score for each word in a sentence [5]. The Predictor-Estimator based QE model is a two-stage model consisting of two types of stacked neural models: first, a neural word prediction model; second, a neural QE model [10]. The Predictor-Estimator architecture uses word prediction as a pre-task for QE. Here, each of the 4 Predictor-Estimator models generates probability or score for each word in a given sentence. And source model generates probability or scores for each word in a given source sentence. These outputs are then stacked together as a new feature, together with source sentences, MT sentences, and word alignment between source and MT sentences, input to a linear sequential model to predict a labeled sequence. The predicted labeled sequence is a sequence with 'OK' or 'BAD' tags, indicating whether the translation for each word is good or

Dataset	*Training*	*Dev*
WMT19+20	20,442	2,000
Common	11,357	988
Similarity	7,500	3,000

Table 2. Data breakdown when using different data selection strategies. *'WMT19+20'* is the combined data sets from WMT19 and WMT20. *'Common'* is the common words based sentence pairs selection. *'Similarity'* is the sentence pairs selected based on Euclidean distance. Please refer to section ***Experiments and Results*** for details about different data selection strategies.

bad (refer to Fig.3 for more details about tags). With this output labeled sequence, we can conduct QE tasks, such as calculating the HTER score (Human-targeted Translation Error Rate).

With the understanding of the model structure, we form our strategy as follows. Predictor-Estimator is the key component in this structure [5]. Thus, our focus has been put on transfer learning these Predictor-Estimator based models, as we think the Predictor-Estimator based models are playing critical roles in determining the performance of the QE tasks. Our baseline model is that trained on English-German language, which, as mentioned before, has presented by ***Lucia, et al 2018*** [11]. The baseline performance is the pre-trained model evaluation on *WMT18* development data set. Again, the data set can be found ***here*** and details of the data set can refer to Table 1. One more thing is worth noticing is that we do not transfer learn all the models used in the ensemble structure, but just the four predictor-estimator based models. It might limit the performance of the final ensemble models after transfer learning. For data set selection used in transfer learning, we have created a new approach to select sentence pairs that are with the most commonly used words and sentence pairs that are with the highest similarity. The details of these methods and results will be discussed in the section of ***Experiments and Results***.

3. Experiments and Results

For transfer learning, we use the ensemble models from the WMT18 submitted work as the baseline [11] and implemented four strategies to fine tuning the parameters for the Predictor-Estimator based models in the original ensemble structure (Fig.4).

The first strategy is to use the entire training data sets combined from WMT19 and WMT20 as the input data. When we apply transfer learning to the pre-trained models, the order of the training data is shuffled. The reason to do the shuffling is to minimize the logic effect in sentence pairs. We also change the dropout rate of the "estimator" model from 0 to 0.1 to overcome the over-fitting issue in the RNN layer. Since the data set has doubled, we change

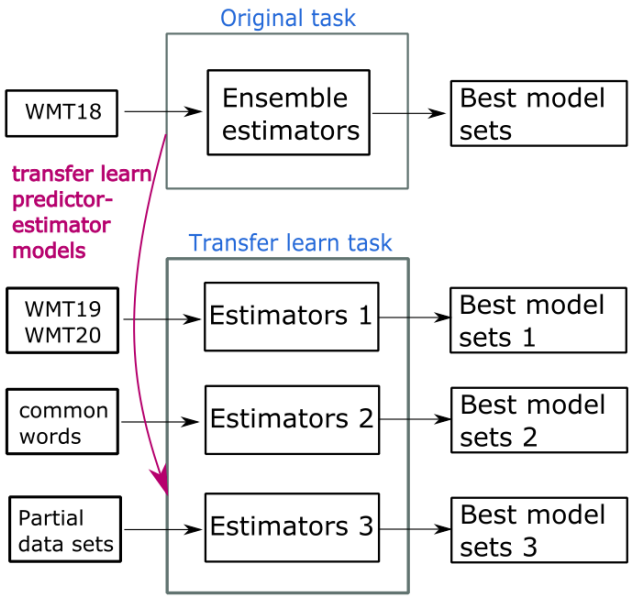

Figure 4. The work flow of transfer learning. The baseline work is trained on WMT18 data set [11] and we use it as baseline and provide three different strategies for transfer learning: one is using all data sets selected from WMT19 and WMT20; the second strategy is select sentences that we with common English words; the third strategy is using partial data sets selected from WMT17, WMT19 and WMT20. Please refer to section ***Experiments and Results*** for more details about different transfer learning and data selection strategies.

Model	*MT tags*	*gap tags*	source tags	*Per-r*	*Spe-r*
NuQE	0.3182	0.1661	0.3030	-	-
APE	0.3754	0.2171	0.3453	-	-
Pred-Est	0.3915	0.2006	0.2933	0.4922	0.5519
Stacked	0.4411	-	-	-	-
Ensemble	0.4298	0.2460	0.3092	0.5214	0.5687

Table 3. Results from baseline reproduction. Here, *'MT tags'* are target tags. *'Per-r'* is Perason correlation and *'Spe-r'* is Spearman correlation.

the training batch size from 64 to 32 and change the validation batch size from 64 to 32, which will speed up the training converge. With a much larger data set and relevant small batch size, we decided to decrease the learning rate from 2e-2 to 1e-2 to avoid noise steps. The combined development sets are used here to help select the best model. With the current training set, the training process took five epochs to reach out to the minimum loss during training.

The second strategy is to use common words from WMT19, 20, with the assumption that better predictions of common words can improve the overall model performance as these words are more likely to show up in real-life prac-

-	Model	*MT tags*	*gap tags*	*Per-r*	*Spe-r*
Transfer1	Pred-Est	0.3739	0.1748	0.4743	0.5191
-	Stacked	0.4248	-	-	-
-	Ensemble	0.4347	0.2188	0.5030	0.5471
Transfer2	Pred-Est	0.3825	0.1810	0.5045	0.5434
-	Stacked	0.4378	-	-	-
-	Ensemble	0.4234	0.2451	0.5228	0.5667
Transfer3	Pred-Est	0.3783	0.1848	0.4911	0.5559
-	Stacked	0.4388	-	-	-
-	Ensemble	0.4210	0.2527	0.5075	0.5691

Table 4. Results from transfer learned models. Here, *'Transfer1'*, *'Transfer2'*, *'Transfer3'* are the results using transfer learning strategy 1, 2 and 3 mentioned in section ***Experiments and Results***, respectively. *'MT tags'* are target tags. *'Per-r'* is Perason correlation and *'Spe-r'* is Spearman correlation.

tice. The common words selection is based on the Oxford 3000, which is a list of 3000 core words that have been chosen based on the frequency in the Oxford English Corpus [8]. This set of words in English was defined as "Common Words" here. In this study, sentence pairs that contributed to the model development process were chosen by a defined ingredient score, which represents the composition of common versus uncommon words for a certain sentence. The ingredient score could be calculated as the frequency of common words over the word count of the whole sentence, which resulted in a fraction between 0 and 1. A large ingredient score means the sentence composition was dominated by common words. A threshold of 0.5 was determined by the grid search to generate sampled data set with an appropriate size.

The third strategy is to use partial data sets selected from WMT17, WMT19, and WMT20 data sets. The selection of the sentences is based on the 2500 sentences that have the least similarity. Euclidean distance is used here to measure the similarity between sentences [6]. Here, Euclidean distance is used as a ranking algorithm, selecting the sentence pairs with top distances. In total, 7500 sentence pairs are selected for training data sets, and 3000 sentence pairs are selected for the development data sets. When conducting transfer learning with the challenge data sets, the order of the training data is shuffled and we decide to maintain almost everything except checkpoint-validation-steps since the size of the training data set is similar. The changes we made are to enrich our training set without adding too many sentences and to tune checkpoint-validation-steps from 0 to 2000 to avoid over-fitting in the invalidation. The training process only takes three epochs to reach out to the minimum loss. The resulted data breakdown is listed in Table 2.

Below, Table 3 shows the performance of our reproduced baseline model. It includes $F1_{mult}$ score for target tags, gap tags, source tag for each model; *Pearson-r* (Pearson correlation) and *Spearman-r* (Spearman correlation) for sentence level scoring and ranking results. Within Table 3 and Table 4, each number is the $F1_{mult}$ score, and it is calculated using the following equation:

$$F1_{mult} = F1_{OK} * F1_{bad} ... (1).$$

where $F1_{OK}$ is the F1 score for words with tags of 'OK' and $F1_{bad}$ is the score for words with tags of 'BAD'. Table 4 summarizes the results using three different transfer learning strategies described in Fig.4.

We decide to use *Spearman-r* and *Pearson-r* to pick the winner. Our model trained using the third strategy wins based on the predictor-estimator model performance. But model trained based on the second strategy wins when we compare Ensemble learners (Table 4). Overall, the baseline model's performance beats all of the transfer learned model performances from our three strategies.

3.1. The effectiveness of transfer learning

Compared with the baseline model self-training process, the transfer learning process of our three different strategies is faster. It only takes two to five epochs to reach the minimum of loss since the smaller size of the data set and strong task of interest in transfer learning will speed up the training process.

The transfer learning could provide comparable model performances and results in general relevant tasks. In our experiments, we introduce the same translation language data sets but with totally different content. Technically, our transfer learned models provide the close result of metrics compared to baseline performance.

3.2. Transfer learning across different data sets

Research has shown that transfer learning will hurt the model with the size of labeled data sets increases [3]. Compared to the results from all the strategies, strategy one has the worst performance. We believe that a large number of training sets hurt the model during transfer learning because transfer learning is not very sensitive to large training sets as it has gained enough general knowledge from previous learning [3]. In addition, larger data sets will bring a specific task of interest. Our baseline model is built on general tasks without a specific interest. In this case, a new transfer learned model with larger data sets may perform worse.

The expectation of the "Common Words" transfer learning strategy was to improve the QE model performance by adding exposures of high-frequency words. However, the final performance did not beat the baseline model. Various reasons may have an impact on the unexpected result. First, the threshold of ingredient score was set to be 0.5, which may still be too low to filter out sentences including numerous words that are considered uncommon. Sec-

ondly, the model performance was evaluated based on the WMT18 data set, which contained IT professional vocabularies and sentences. A refined IT-specific definition of "Common Word" will provide more predicting power than the approach using Oxford 3000.

By learning from previous experiments, the model performance of strategy three still could not beat the baseline model performance. Here, we enrich the content of the data, which generalizes the interest of this task. Transfer learning is supposed to be doing well. Then we decide to check predictor-estimator model performance on its corresponding development set. When we applied estimator model comparison on its corresponding development set, our model performances of second and third strategies have similar performance compared to the baseline model. Thus, the main reason we think is that other parts such as NuQe and APE could not match well with the new transfer learned P-E models.

4. Conclusion and Future Work

In this work, we researched the QE system for machine-translated sentences and reproduced the baseline model for word level and sentence level estimation using the WMT18 English-German data set. Then, we tried three transfer learning methods to explore the potentials to improve the baseline model. Our strategies were focused on data augmentations to improve model performances: the entire WMT19+20 data, the sampled sentences based on common words, and sentence pairs based on the sentence level similarity. Our approach is concise, efficient, and open source. Although the transfer learned model performance did not beat the baseline which is one of the winning submissions for WMT18, our transfer learned results are still useful in providing insights to understand that larger data set may not help learn. These results can be utilized as a good preliminary work to support future research in the pre-trained and self-train models or related fields.

Future work regarding transfer learning may be conducted from various perspectives. One interesting topic would be to generalize the current model to Multi-Language tasks instead of English-German only. Approaches such as Multi-Bert which provides word embedding across different languages could be the replacement of model input. Besides, Multi-Task Learning is a popular direction for Quality Estimation related problems. By learning several tasks simultaneously, the model could optimize multiple performance metrics at the same time. One example is to consider the model's performance on multiple language data sets through the learning process.

On the other hand, we believe that transfer learned Predictor-Estimator models could not align well with non-transfer learned NuQe and APE models. If we enrich the content of data in NuQe and APE at the same time when conducting transfer learning, the overall performance of the model generated by our strategies will be better than in current experiments. Based on the above information, we could hypothesize that if we enrich input data such as source sentences, MT sentences, or Word alignments to the linear sequential model, the performance of the transfer learned model can be improved.

Since we have been successfully applied transfer learning using different strategies in this project, it provides us insight into how to leverage transfer learning in new data. The fine-tuned transfer learning models can perform as well as the self-trained model and can save tons of time in the training process. In future machine translation applications, transfer learning will play a critical role.

Name	***Lit***	***Baseline***	***TL***	***Exp***	***Writing***
MZ	✓	✓	-	✓	✓
HY	✓	✓	-	✓	✓
BY	✓	-	✓	-	✓
JZ	✓	-	✓	✓	✓

Table 5. Work division between team members

5. Work Division

Team members are all actively contributing to the final project deliverable. Specifically, all team members conduct the literature review (*Lit*). HY and MZ reproduced the baseline model frame work (*Baseline*) including recreating all the coding files, data cleaning and conducting baseline evaluation experiments for transfer learning. HY and MZ also designed the experiments (*Exp*). HY, MZ, JZ contributed to most of the writing (*Writing*) of final report. BY contributed a little in writing the data selection for transfer learning. BY and JZ conduct work for searching transfer learning (*TL*) related methods. JZ conduct data extraction for transfer learning and running transfer learning related experiments (*Exp*). Please refer to table ***Work division between team members*** for a high level summary.

References

[1] Chris Hokamp Andre F. T. Martins, Ramon Astudillo and Fabio N. Kepler. Unbabel's participation in the wmt16 word-level translation quality estimation shared task, 2016. 1

[2] Fabio N. Kepler Ramon Astudillo Chris Hokamp Andre F. T. Martins, Marcin Junczys-Dowmunt and Roman Grundkiewicz. 3

[3] Tsung-Yi Lin Yin Cui Hanxiao Liu Ekin D. Cubuk Barret Zoph, Golnaz Ghiasi and Quoc V. Le. Rethinking pre-training and self-training, 2020. 5

[4] Marcos Treviso Miguel Vera Fábio Kepler, Jonay Trénous and André F. T. Martins. 2, 3

[5] Marcos Treviso Miguel Vera António Góis M. Amin Farajian António Lopes Fábio Kepler, Jonay Trénous and André F. T. Martins. Unbabel's participation in the wmt19 translation quality estimation shared task, 2019. 1, 3, 4

[6] Juri Ranieri Ivan Dokmanic, Reza Parhizkar and Martin Vetterli. 5

[7] Shigehiko Schamoni Julia Kreutzer and Stefan Riezler. Quality estimation from scratch (quetch): Deep learning for word-level translation quality estimation, 2015. 1

[8] James Milton Julie Moore, Marlise Horst and Paul Nation. 5

[9] Hyun Kim and Jong-Hyeok Lee. A recurrent neural networks approach for estimating the quality of machine translation output, 2016. 1

[10] Hyun Kim and Jong-Hyeok Lee. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation, 2017. 1, 3

[11] Blain Frederic Fernandez Ramon Specia Lucia, Logacheva Varvara and Martins André. 1, 2, 3, 4