Estimation vs Metrics: is QE Useful for MT Model Selection?

Anna Zaretskaya **José Conceição** **Frederick Bane**
TransPerfect
Passeig de Gràcia, 11, Esc B 5-2
08007 Barcelona, Spain
{azaretskaya, jconceicao, fbane}@translations.com

Abstract

This paper presents a case study of applying machine translation quality estimation (QE) for the purpose of machine translation (MT) engine selection. The goal is to understand how well the QE predictions correlate with several MT evaluation metrics (automatic and human). Our findings show that our industry-level QE system is not reliable enough for MT selection when the MT systems have similar performance. We suggest that QE can be used with more success for other tasks relevant for translation industry such as risk prevention.

1 Introduction

Machine translation quality estimation (QE) is a technique for predicting machine translation (MT) quality (Specia et al., 2009). As MT becomes the dominant tool in the translation industry, accurate estimation of the quality of MT output would be of great benefit to many business concerns such as budget allocation for human post-editing, estimating the usefulness of the MT output for gisting purposes, and selecting the best MT system out of a selection of systems. In addition to that, a reliable QE model would also help linguists make more efficient use of their time.

As opposed to MT evaluation, where MT output is compared to one or several human reference translations, QE attempts to perform the much more challenging task of predicting MT quality in the absence of a reference translation. QE can be performed on a word, sentence or document level, and the output of a QE system is typically a score that is intended to correlate with a certain automatic or human MT evaluation metric.

In this paper we present a case study where QE models were applied for the purpose of ranking different MT engines for a given document or text corpus. Our approach is based on obtaining segment-level QE scores for all segments in the document/corpus and using the average to select the best MT system. We use two QE systems to score the output of different MT engines and compare the results of the QE model with several automatic MT evaluation metrics, which include the post-editing distance (PED), HTER, BLEU score (Papineni et al., 2002), and weighted and unweighted sentence embedding similarity, as well as with human MT evaluation scores. We conclude the paper by reflecting on the usefulness of QE for MT engine selection, possible improvements to QE, and the limitations of our model and method.

2 Background and Motivation

The main motivation of exploring the QE method for MT model selection is the fact that we are often faced with a scenario where we need to choose the best MT system without a reference translation. The only existing method for this is involving human evaluators, which is, however, quite costly and requires more time. We are looking for a more cost-efficient and fast (almost immediate) way of deciding which MT system to use.

QE is a very well explored topic in Natural Language Processing given that predicting MT quality has clear practical benefits (Specia et al., 2018). Multiple QE frameworks have been developed, some of which are open-source: QuEst++ (Specia et al., 2015), POSTECH (Kim et al., 2017),

QEBrain (Wang et al., 2018), OpenKiwi (Kepler et al., 2019), YiSi (Lo, 2019) and others. Details about state-of-the-art QE tools are presented in detail in the corresponding WMT2019 shared task (Fonseca et al., 2019).

Despite the extensive research interest in QE, there is less information on how useful it actually is in specific commercial workflows. For example, Shterionov et al. (2019) compare the performance of various QE models using specific user business metrics, as well as implementation and computation cost. They demonstrate that the system with the highest performance can be also the most computationally expensive and simpler, faster systems can provide satisfactory results. More aspects of applying QE in commercial settings are discussed in (de Souza et al., 2015) and (Astudillo et al., 2018).

We believe that evaluation of the performance of any NLP system must firstly take into account the end use of the system. In our case, the goal is to be able to automatically select the best MT engine out of two or more engines on a segment level in scenarios where a reference translation is not available. It is relevant to mention that our goal is not to fully replace automatic MT evaluation metrics; the findings from Task 3 of the shared task on QE (Fonseca et al., 2019) confirm that this is still a challenge. Rather, the objective of this study is to investigate whether our QE systems are reliable enough to be used to select the best among multiple MT engines.

Segment-level QE is typically evaluated by calculating the correlation of the QE predictions with human judgement or one or several MT evaluation metrics, most commonly with HTER (Snover et al., 2006a), using the Root Mean Square Error (RMSE) and/or Mean Absolute Error (MAE) (Specia et al., 2018). Apart from that, Avramidis et al. (2018) describe a more fine-grained, linguistically-informed evaluation method which enables greater understanding of the behaviour of the QE system.

For this study, in addition to the standard metrics, we utilize the metrics that correspond to our business goals. One of the most important metrics for us is the post-editing distance (PED), a standard MT quality metric used at the company and is the current industry standard. Similarly to HTER, PED represents post-editing effort in terms of the number of editing operations, but it is character-based (while HTER is word-based) and therefore more accurately reflects the effort expended in editing. Another important metric for us is the BLEU score (Papineni et al., 2002), which is used mostly for MT development in order to measure improvement over a baseline.

There are many valid criticisms of automatic MT evaluation metrics such as BLEU (Callison-Burch et al., 2006), one of the most salient of which is the fact that they require one or more reference translations against which MT output is compared. However, a given sentence can have multiple correct translations depending on a certain context and end use, and for this reason reference-based metrics cannot always cover the entire space of valid translations for the sentence. For this reason, we also include human MT evaluation - direct assessment of the MT output by linguists, which is not reference-dependant. In addition, we experiment with text similarity metrics (Chan and Ng, 2008). In particular, we use word embedding similarity in order to reflect how semantically close the MT translation is to the reference translation. We produced sentence embeddings for the MT output and the reference translation, and calculated the cosine distances between these embeddings. Cosine distances between sentence embeddings capture how closely the meanings of two sentences correspond in high-dimensional vector space, and as such are less sensitive to the substitution of similar words in alternative translations. While this latter metric is insufficient on its own as a measure of translation quality (due to its insensitivity to word-order, among other reasons), we hypothesize that it may be a useful auxiliary metric.

3 Methodology

Our primary research goal was to investigate how well the scores from QE systems correlate with commonly-used metrics for evaluating translation quality, and based on these results understand how useful QE is for MT model selection in cases when a reference translation is not available. As a secondary goal, we also studied the impact of content domain on these correlations. Domain is known to be a highly significant factor in the performance of MT engines, and we hypothesized that this would also be true of QE systems. Therefore, for this study we used two QE models: one in a general domain (QE-gen) and the other in the domain of life sciences (QE-domain). Below we present im-

plementation details about these models.

3.1 QE Systems

The QE models used to perform these experiments were implemented using the OpenKiwi framework (Kepler et al., 2019). The framework was chosen as it was the foundation of the winning systems of the word-, sentence-, and document-level tasks of the WMT 2019 shared task on QE, and furthermore because of its adoption as the baseline system for this task (Fonseca et al., 2019).

When it comes to the architecture, we chose to use the Predictor-Estimator (Kim et al., 2017), a two-phase, end-to-end neural QE model which had the most noteworthy benchmarks of all OpenKiwi's available architectures (excluding ensembles and stacks). The Predictor-Estimator architecture attempts to overcome challenges faced by previous architectures, such as a shortage of QE data and dependence on hand-engineered features to capture the complex relationships between feature sets and QE annotations.

This architecture uses word prediction as a pretask to boost performance and reduce the amount of QE data needed to achieve state-of-the-art results. This task takes in source and target sentences, masks a target word at random, and then attempts to predict the masked word. Word prediction uses a bidirectional long short-term memory (LSTM) to encode the source and two unidirectional LSTMs to process the target: LSTM-L2R (left to right) and LSTM-R2L (right to left) (Kepler et al., 2019). These LSTMs are trained using a large parallel corpus. This structure allows the use of both left and right target context to generate predictions of the masked word.

Before diving into detailed descriptions of the models, it is worth noting that the systems used in scientific research are normally ensembles or stacks of different architectures, which typically outperform individual stand-alone systems. However, at this stage we think the difference is not substantial enough to justify the increased costs of training several models for each language pair instead of one, which can skyrocket when taking into consideration the number of language pairs that our company handles. Both QE models (the domain and the generic one) used the same word predictor model, built from a large generic parallel corpus. The primary difference is due to the different text types from which the data were sampled: QE-domain model was trained exclusively on texts from the Life Sciences domain and QE-gen was trained on a mixed corpus. In both cases, the training data was compiled from previously post-edited projects. Table 1 shoes the training corpora size used for each of the models.

The first step in the pre-processing pipeline was to query our SQLite database, specifying our language and other settings such as the maximum number of tokens per sentence. We then refined the data using the langdetect python package to filter out any rows that weren't flagged with the language pair we had specified. To generate the OK/BAD tags (the tags marking whether a specific word is correct or wrong in the translation), we relied on the industry standard TERCOM tool (Snover et al., 2006b). For each token in each sentence, if the token is present in the target sentence, the token is labeled OK; if it was deleted or modified during PE, the token is labeled BAD. Insertions are ignored. At the end, sentences are updated such that only the ones without any error in identifying the tags are kept. Both models achieved industry-standard F1-mult scores. F1-mult is a word-level prediction score that evaluates the performance of identifying correct and incorrect words in the translation (Table 1).

	QE-gen	**QE-domain**
F1-mult	55.73	57.85
Test corpus	2893	1998
Training corpus	134438	92341

Table 1: Training and testing corpus size in number of sentence pairs and the F1-mult score of the two QE models.

3.2 Experiments

Equipped with these two models, we conducted two separate experiments, one in the general domain and one in the life sciences domain. In the first, we obtained translations for the generic data set from two freely available MT engines, Google and Bing. The QE-gen model was used to predict the quality of these translations, then these scores were compared with several metrics for evaluating translation quality. The data obtained from these comparisons were considered a baseline by which to judge the performance of the two QE models on domain-specific content in the second experiment.

In the second experiment, we used a dataset of life sciences content to compare the performance

of the QE-gen and QE-domain models. In the first experiment we found that the Bing and Google MT engines performed quite similarly in terms of the quality of their output. Thus, for this experiment we also used a specialized proprietary life sciences MT engine, which we expected to perform significantly better than the two more general engines. Translations were obtained from all three engines, and these translations were scored by the QE-domain and QE-gen models. The resulting QE scores were then compared to the same MT evaluation metrics.

In addition to measuring how well our QE models correlate with MT quality metrics, we also calculated the probabilities of the QE models to correctly identify the best MT engine out of several.

3.3 Test Data

We used two sets of data for the evaluation. The first set contained 1756 sentences translated from English into Spanish by professional translators from the corporate communication domain. We selected these texts because they have a general style and do not have any specific or technical terminology. The average source sentence length was 17.91 words.

The second data set contained 2048 sentences from the Life Sciences domain and contained texts with highly specialized terminology and style. The average sentence length in this data set was 14.29 words. Both data sets were cleaned to remove sentences with less than four and more than 200 tokens as well as any sentences where the MT outputs of the engines were identical.

3.4 Evaluation

Each QE system (QE-gen and QE-domain) was evaluated based on

- the correlation of the QE scores and PED (Pearson's r);
- the correlation of the QE scores and BLEU (Pearson's r);
- the correlation of the QE scores and the two sentence similarity metrics (Pearson's r);
- the RMSE (root mean square error) for HTER;
- the MAE (mean absolute error) for HTER;
- the percentage of sentences where the QE model correctly selected the best MT engine based on each of the quality metrics;
- correlation with human assessment of the MT quality.

PED was calculated using the Levenshtein distance algorithm at the character level and normalized based on the length of the strings. HTER scores were calculated as explained in (Snover et al., 2006a). BLEU scores were assigned using NLTK's built-in BLEU score function. The text similarity metrics were calculated as the cosine distances between the weighted and unweighted sentence embeddings of the MT output and the human translation. As such, lower values indicate more similar sentences. Unweighted sentence embeddings were calculated as a simple mean of Word2Vec word embeddings for each word in the sentence, while weighted sentence embeddings were calculated by averaging the word embeddings after weighting them based on the inverse frequency of the word in the Word2Vec[1] training corpus. The Scipy and Numpy python libraries were used to perform data analysis, and the Pearson's correlation coefficient (PCC) was used to assess correlation between the QE scores and our other translation quality metrics.

For the human evaluation we have used 200 sentences from each dataset, which were evaluated by two different annotators on a 1 to 100 scale. During the evaluation, the reference translations of the segments were not provided. The Human judgement scores were then averaged between the two annotators. Then, we followed the procedure described in (Ma et al., 2019) to calculate the Kendall's τ scores that show the correlation between the QE scores and the human judgment. It has to be noted that we removed all the instances of ties in human judgment, i.e. all the segments where the MT engines were assigned the same average human score. After removing all the human judgment ties, we ended up with 134 segments in each of the datasets. As to the ties in the QE scores, these were penalized, meaning that we counted as *Discordant* the segments where the predicted QE scores for different MT systems were equal (and the human scores were not).

[1] https://arxiv.org/pdf/1310.4546.pdf

4 Results

We evaluated the two QE models on the corresponding data sets in terms of the model performance. Table 2 shows the Pearson's correlation results with the automatic MT evaluation metrics. In the Generic Use Case, we used QE-gen and the generic data set. Here, we compare the results only for the two generic (not customized) MT systems. In the Domain use case, we used QE-domain and the Life Sciences data set. In the Mixed Use Case, we used QE-gen and the Life Sciences data set. In the latter two cases we also consider the results of the domain specific MT system trained for life sciences content. The Mixed case allows us to compare the performance of a domain-specific QE system with that of a generic QE system. Similar results would suggest no clear benefit from training different systems for each genre of content.

		Google	**Bing**	**LifeSci**
Generic Case	PED	0.301	0.278	
	HTER	0.148	0.089	
	BLEU	-0.308	-0.271	
	Sim1	0.296	0.203	
	Sim2	0.198	0.141	
Domain Case	PED	0.284	0.199	0.127
	HTER	0.308	0.273	0.135
	BLEU	-0.324	-0.302	-0.195
	Sim1	0.315	0.308	0.184
	Sim2	0.180	0.166	0.118
Mixed Case	PED	0.261	0.182	0.125
	HTER	0.280	0.276	0.138
	BLEU	-0.269	-0.290	-0.188
	Sim1	0.245	0.269	0.175
	Sim2	0.159	0.172	0.090

Table 2: Pearson's correlation results between the predicted sentence-level QE score and the particular MT metrics. Sim1 refers to unweighted sentence similarity, while Sim2 refers to weighted sentence similarity.

In general, we found only weak correlation with most of the metrics and in some cases almost no correlation at all. While the F1-mult scores indicated that our QE models achieved industry-level performance, the poor correlations with the evaluation metrics were unexpected. Out of all the metrics considered, the highest correlation observed was for the BLEU score. Interestingly, the correlation with HTER was particularly weak (practically no correlation) in the generic case, but stronger for life sciences domain content. When it comes to the word embedding similarity, using unweighted embeddings proved to yield a stronger correlation with QE than weighted embeddings. This may be partially explained by the fact that our weighted embeddings distinguish words in terms of their frequency, while QE systems and unweighted word embeddings treat all words equally.

Table 3 shows the RMSE and the MAE scores for the predicted HTER. Based on RMSE, the predicted HTER scores differ from the actual HTER scores by about 5 percentage points, while based on MAE calculation the difference is about 3 percentage points.

		QE-domain	**QE-gen**
RMSE↓	Google	5.186	4.949
	Bing	5.190	5.156
	LifeSci	5.450	5.373
MAE↓	Google	3.574	3.437
	Bing	3.606	3.541
	LifeSci	3.656	3.732

Table 3: RMSE and MAE of each of the QE models applied to the output of the three MT engines.

Finally, the correlation with human judgment in terms of Kendall's τ is also weak or non-existing. For the generic dataset, the τ score was equal to 0.119 (slightly better than random) while for the in-domain dataset the τ score was equal to –0.059 (practically random). Note that the τ score can take value from –1 to 1.

The observations indicate how well our QE systems perform and how similar their behavior is to the various metrics. However, we want to understand whether their performance level is sufficient to be able to replace MT evaluation metrics for the purpose of engine selection. Therefore, we also provide a comparison of the average metrics scores for the different MT engines with the average QE scores (Tables 4 and 5).

As can be seen from these results, the performance of the two generic MT systems (Bing and Google) was very similar according to all the metrics and also the average QE scores. While Google and Bing score better according to the automatic metrics, human evaluation ranked Google first, and the QE system is in line with the human score. In general, though, the differences between the two were negligible.

On the other hand, the tendency changes when the Life Sciences MT engine comes into the pic-

	Google	Bing
PED ↓	0.321	0.311
HTER ↓	0.468	0.459
BLEU ↑	0.682	0.699
Sim1 ↓	0.081	0.079
Sim2 ↓	0.010	0.010
Human ↑	83.5	82.6
QE-gen ↓	0.312	0.325

Table 4: Average values for the automatic and human MT evaluation metrics compared to average QE score QE-gen on the Generic data set.

	Google	Bing	LifeSci
PED ↓	0.296	0.282	0.183
HTER ↓	0.413	0.397	0.253
BLEU ↑	0.328	0.350	0.561
Sim1 ↓	0.055	0.052	0.027
Sim2 ↓	0.006	0.005	0.003
Human ↑	83.6	85.3	88.5
QE ↓	0.396	0.392	0.372

Table 5: Average values for the automatic and human MT evaluation metrics compared to average QE score of QE-gen (for Google and Bing MT systems) and QE-domain (for LifeSci MT system) on the Life Sciences data set.

ture (Table 5); its performance is significantly higher according to all the metrics, and therefore the QE system also correctly identifies it as the best engine out of three (although the difference is rather small). This is also illustrated in Figure 1, which shows the distribution of PED scores and QE scores for the three engines on the Life Sciences data set. These results suggest that QE systems are more likely to choose the best model in cases where one MT engine clearly outperforms

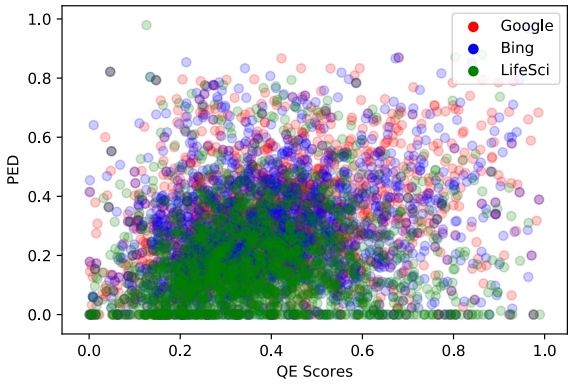

Figure 1: Distributions of QE scores and PED for the three engines on life sciences data. Note the "squashing" of the distribution for the LifeSci engine

the others. This conclusion is also in line with the findings of the WMT19 Metrics Shared Task (Ma et al., 2019), which conclude that the metrics and QE tasks become more challenging when comparing multiple strong systems with similar performance as opposed to scenarios where the performance level of the systems is more varied.

	Generic	Domain	Mixed
PED	53.4%	44.8%	47.8%
HTER	41.0%	44.5%	49.0%
BLEU	51.6%	39.5%	45.1%
Sim1	51.0%	43.5%	46.6%
Sim2	47.7%	43.0%	46.2%

Table 6: Percentage of cases where QE correctly selected the best MT engine based on each of the automatic MT evaluation metrics. Note that for the generic case only 2 MT engines were used, so the results are essentially random.

Finally, we consider the percentage of cases (segments) where the QE systems correctly identified the best engine based on each of the metrics (Table 6). In the Generic case (two generic MT engines used on generic data), the results are practically random. In the Domain and Mixed scenarios, where three MT engines were used, the best engine is correctly identified in about 50% of the cases. These results are noticeably better than a random guess (which would be correct 33% of the time), but are not sufficient to meet the standard of usability in our workflow.

In summary, we observe mediocre results. Based on the weak correlation with the MT evaluation metrics and the human judgment we can conclude that our QE systems do not perform well enough in order to be used on a sentence level. On the other hand, when considering the average QE scores across the entire test set, we do see that a superior MT engine does tend to have lower average QE scores. This suggests that, at the document level, our QE models might do better at identifying the best MT engine in scenarios where the performance of the MT engines is significantly different - this is still to be confirmed by further studies.

We did not observe a significant gap in performance between the QE model trained entirely on life sciences data and the generic model when applied to life sciences content. Indeed, despite the fact that the QE-domain achieved superior a F1-mult score, this model performed worse than the QE-gen model at predicting the best sentence on every metric.

5 Discussion and Future Work

In this paper we explored how well industry-standard QE models correlate with traditional measures of MT quality, both in a specific domain and in a general domain scenario. Our goal was to establish if these models can be used for automatic MT engine ranking. Our models used the OpenKiwi framework and achieved F1-mult scores similar to currently reported scores for similar single models. However, we failed to find strong evidence that these scores translate reliably into predictions of which MT engine's translation has better quality.

While the results we obtained are better than random guessing, we can conclude that QE in its current state can be fruitfully applied for MT model selection only in very specific scenarios, namely when the given MT models are known or expected to differ significantly in performance. Nevertheless, it is encouraging that there is some observable correlation between QE and our various metrics, and that our QE systems did show a tendency to choose the best model when there was a clearly superior choice. Indeed, in real production scenarios, there is no risk of choosing a slightly worse MT system when its performance is comparable to the other candidates, while it is more important to filter out the systems with significantly lower performance. In addition, we suggest that a very useful application of QE to explore is risk prevention: instead of selecting the best MT system out of several, we would be able to predict with a high degree of confidence that the performance of an MT system is significantly lower than average. This is one of the directions we are planning to explore in future studies.

One of the immediate steps in our research will be qualitative analysis of the data, especially the of the segments where a significant discrepancy was observed between the human evaluation scores and the QE scores. We hope to obtain an more profound understanding of the data and the reasons for the weak correlation.

When it comes to the actual performance of our QE system, the question becomes, how can we improve ours so that it may be more useful in the future? The first idea that presents itself is to reduce the class imbalance during the training stage. In the dataset on which we trained our QE models, OK tags outnumbered BAD tags by a factor of nearly 10:1. We hypothesize that the performance of the classifier may improve if we better balance the examples of the OK and BAD classes. One way to accomplish this goal is through the use of synthetic training data. In addition to real examples, we could create additional examples by replacing words randomly with other words from the vocabulary (either sampled uniformly or weighted based on the frequency that the word is associated with a BAD tag), thereby increasing the number of BAD tags the system sees during training.

Another possible way to improve the performance of QE models is through adversarial training. Using an architecture similar to a GAN, we could train a generator to create predictions for each word in a sentence, and simultaneously use the output of this system and human-annotated sentences to train a discriminator to distinguish model-generated output from human-annotated sentences. At this time we are not aware of any study which attempts to implement these methods for QE.

One important observation about the QE system's performance that we can draw from this study is that contrary to our expectations, there was no boost in performance compared with the generic model when an in-domain QE model was used on in-domain content. One reason for this might be that the QE-gen model was exposed to more data (including all the data used to train the QE-domain model), and so it may have developed a more sophisticated and robust language model than its counterpart trained on only a subset of those data. Another possibility is that domain simply does not play as significant a role in QE modeling as it does in more complex generative tasks like translation. In any case, it is a rather positive finding, as it proves that there is no need to train a QE model for each domain and training one generic model on a corpus that contains data from different domains is sufficient.

References

Astudillo, Ramón, João Graça, and André Martins, editors. 2018. *Proceedings of the AMTA 2018 Workshop on Translation Quality Estimation and Automatic Post-Editing*, Boston, MA, March. Association for Machine Translation in the Americas.

Avramidis, Eleftherios, Vivien Macketanz, Arle Lommel, and Hans Uszkoreit. 2018. Fine-grained evaluation of quality estimation for machine translation based on a linguistically motivated test suite. In *Proceedings of the AMTA 2018 Workshop on Transla-*

tion Quality Estimation and Automatic Post-Editing, pages 243–248, Boston, MA, March. Association for Machine Translation in the Americas.

Callison-Burch, Chris, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, April. Association for Computational Linguistics.

Chan, Yee Seng and Hwee Tou Ng. 2008. MAXSIM: A maximum similarity metric for machine translation evaluation. In *Proceedings of ACL-08: HLT*, pages 55–62, Columbus, Ohio, June. Association for Computational Linguistics.

de Souza, Jose G. C., Marcello Federico, and Hassan Sawaf. 2015. Mt quality estimation for e-commerce data. In *Proceedings of MT Summit XV, vol. 2: Users' Track*, pages 20–29, Maiami, Florida.

Fonseca, Erick, Lisa Yankovskaya, AndrÃ© F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–12, Florence, Italy, August. Association for Computational Linguistics.

Kepler, Fabio, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy, July. Association for Computational Linguistics.

Kim, Hyun, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568, Copenhagen, Denmark, September. Association for Computational Linguistics.

Lo, Chi-kiu. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy, August. Association for Computational Linguistics.

Ma, Qingsong, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy, August. Association for Computational Linguistics.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shterionov, Dimitar, Félix do Carmo, Joss Moorkens, Eric Paquin, Dag Schmidtke, Declan Groves, and Andy Way. 2019. When less is more in neural quality estimation of machine translation. an industry case study. In Forcada, Mikel L., Andy Way, John Tinsley, Dimitar Shterionov, Celia Rico, and Federico Gaspari, editors, *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks, MTSummit 2019, Dublin, Ireland, August 19-23, 2019*, pages 228–235. European Association for Machine Translation.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006a. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006b. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas, 2006.*

Specia, Lucia, Nicola Cancedda, Marc Dymetman, Marco Turchi, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *Proceedings of the 13th Annual Conference of the European Association for Machine Translation (EAMT 2009)*, pages 28–35, Barcelona, Spain.

Specia, Lucia, Gustavo Henrique Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with QuEst++. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 115–120, Beijing, China.

Specia, L., C. Scarton, and G. H. Paetzold. 2018. *Quality Estimation for Machine Translation*, volume 11(1) of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers.

Wang, Jiayi, Kai Fan, Bo Li, Fengming Zhou, Boxing Chen, Yangbin Shi, and Luo Si. 2018. Alibaba submission for WMT18 quality estimation task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 809–815, Belgium, Brussels, October. Association for Computational Linguistics.