

OpenKiwi: An Open Source Framework for Quality Estimation

Fábio Kepler
Unbabel

Jonay Trénous
Unbabel

Marcos Treviso*
Instituto de Telecomunicações

Miguel Vera
Unbabel

André F. T. Martins
Unbabel

{kepler, sony, miguel.vera, andre.martins}@unbabel.com
marcosvtreviso@gmail.com

Abstract

We introduce OpenKiwi, a Pytorch-based open source framework for translation quality estimation. OpenKiwi supports training and testing of word-level and sentence-level quality estimation systems, implementing the winning systems of the WMT 2015–18 quality estimation campaigns. We benchmark OpenKiwi on two datasets from WMT 2018 (English-German SMT and NMT), yielding state-of-the-art performance on the word-level tasks and near state-of-the-art in the sentence-level tasks.

1 Introduction

Quality estimation (QE) provides the missing link between machine and human translation: its goal is to evaluate a translation system’s quality without access to reference translations (Specia et al., 2018b). Among its potential usages are: informing an end user about the reliability of automatically translated content; deciding if a translation is ready for publishing or if it requires human post-editing; and highlighting the words that need to be post-edited.

While there has been tremendous progress in QE in the last years (Martins et al., 2016, 2017; Kim et al., 2017; Wang et al., 2018), the ability of researchers to reproduce state-of-the-art systems has been hampered by the fact that these are either based on complex ensemble systems, complicated architectures, or require not well-documented pre-training and fine-tuning of some components. Existing open-source frameworks such as QuEST++ (Specia et al., 2015), Marmot (Logacheva et al., 2016), or DeepQuest (Ive et al., 2018), while helpful, are currently behind the recent best systems in WMT QE shared tasks. To address the shortcoming above, this paper presents **OpenKiwi**¹ a new

open-source framework for QE that implements the best QE systems from WMT 2015-18 shared tasks, making it easy to combine and modify their key components, while experimenting under the same framework.

Training a new OpenKiwi model is as simple as running the following command line:

```
$ python kiwi train --config
    config.yml
```

where `config.yml` is a configuration file with training and model options. A stacked combination of these models yields state-of-the-art results for word-level QE on the widely used English-German SMT and NMT datasets. For ease of use from the research community, we made our code-base clean and modular, with detailed documentation and high test coverage.

The main features of OpenKiwi are:

- Implementation of five QE systems: QUETCH (Kreutzer et al., 2015), NUQE (Martins et al., 2016, 2017), Predictor-Estimator (Kim et al., 2017; Wang et al., 2018), APE-QE (Martins et al., 2017), and a stacked ensemble with a linear system (Martins et al., 2016, 2017);
- Implementation in Python 3 using PyTorch as the deep learning framework;
- Easy to use API: can be imported as a package in other projects or run from the command line;
- Ability to train new QE models on new data;
- Ability to run pre-trained QE models on data from the WMT 2018 campaign;
- Easy to track and reproduce experiments via YAML configuration files;

* Work done during an internship at Unbabel in 2018.

¹<https://unbabel.github.io/OpenKiwi>.

- Open-source license (Affero GPL).

This project is hosted at <https://github.com/Unbabel/OpenKiwi>. We welcome and encourage contributions from the research community.²

2 Quality Estimation

The goal of **word-level QE** (Figure 1) is to assign quality labels (OK or BAD) to each *machine-translated word*, as well as to *gaps* between words (to account for context that needs to be inserted), and *source words* (to denote words in the original sentence that have been mistranslated or omitted in the target). In the last years, the most accurate systems that have been developed for this task combine linear and neural models (Kreutzer et al., 2015; Martins et al., 2016), use automatic post-editing as an intermediate step (Martins et al., 2017), or develop specialized neural architectures (Kim et al., 2017; Wang et al., 2018).

Sentence-level QE, on the other hand, aims to predict the quality of the whole translated sentence, for example based on the time it takes for a human to post-edit it, or on how many edit operations are required to fix it (Specia et al., 2018b). The most successful approaches to sentence-level QE to date are based on conversions from word-level predictions (Martins et al., 2017) or joint training with multi-task learning (Kim et al., 2017; Wang et al., 2018).

3 Datasets

To benchmark OpenKiwi, we use the following datasets from the WMT 2018 quality estimation shared task, all English-German (En-De):

- Two quality estimation datasets of sentence triplets, each consisting of a source sentence (SRC), its machine translation (MT) and a human post-edition (PE) of the machine translation. The data also contains word-level quality labels and sentence-level scores that are obtained from the post-editions using TERCOM (Snover et al., 2006).
 - A larger dataset of 26,273 training and 1,000 development triplets, where the MT is generated by a phrase-based statistical machine translation (SMT).

- A smaller dataset of 13,442 training and 1,000 development triplets, where the MT is generated by a neural machine translations system (NMT).

- A corpus of 526,368 artificially generated sentence triplets, obtained by first cross-entropy filtering a much larger monolingual corpus for in-domain sentences, then using round-trip translation and a final stratified sampling step.
- A parallel dataset of 3,396,364 in-domain sentences used for pre-training of the predictor-estimator model.

4 Implemented Systems

OpenKiwi implements five popular systems that have been proposed in the last years, which we now describe briefly.

QUETCH. QUETCH (Kreutzer et al., 2015) is designed as a multilayer perceptron with one hidden layer, non-linear *tanh* activation functions and a lookup-table layer mapping words to continuous dense vectors. For each position in the MT, a window of fixed size surrounding that position, as well as a windowed representation of aligned words from the source language are concatenated as model input. The output layer scores OK/BAD probabilities for each word with a softmax activation. The model is trained independently to predict source tags, gap tags, and target tags. QUETCH is a very simple model and does not rely on any kind of external auxiliary data for training, only the shared task data sets.

NuQE. NuQE is a neural quality estimation system proposed by Martins et al. (2016). Its architecture consists of a lookup layer containing embeddings for target words and their source-aligned words.³ These embeddings are concatenated and fed into two consecutive sets of two feed-forward layers and a bi-directional GRU layer. The output contains a softmax layer that produces the final OK/BAD decisions. Gap tags are handled separately from target tags, requiring standalone models to be trained. Likewise for source tags. NuQE is a blackbox system, meaning we train it with the

²See <https://unbabel.github.io/OpenKiwi/contributing.html> for instructions for contributors.

³In the experiments, we use the alignments provided by the shared task organizers, which are computed with `fast_align` (Dyer et al., 2013).

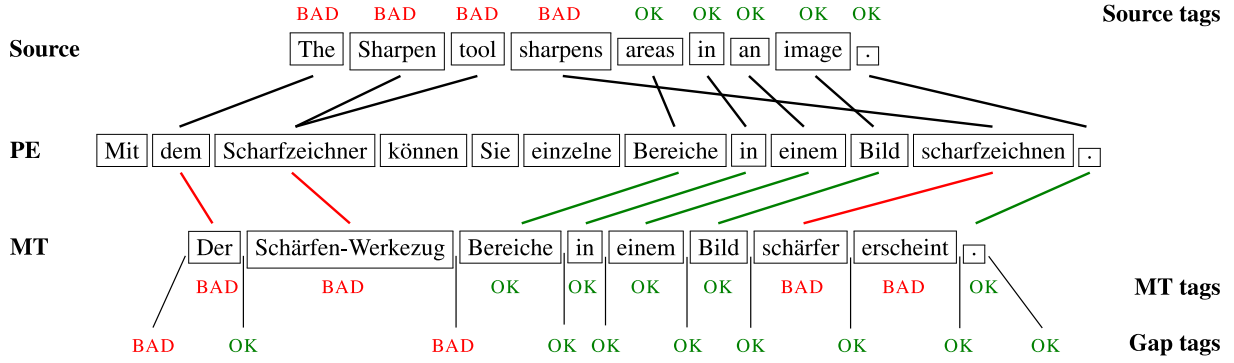


Figure 1: Example from the WMT 2018 word-level QE training set. Shown are the English source sentence (top), the German machine translated text (bottom), and its manual post-edition (middle). We show also the three types of word-level quality tags: MT tags account for words that are replaced or deleted, gap tags account for words that need to be inserted, and source tags indicate what are the source words that were omitted or mistranslated. For this example, the HTER sentence-level score (number of edit operations to produce PE from MT normalized by the length of PE) is $8/12 = 66.7\%$, corresponding to 4 insertions, 1 deletion, and 3 replacements out of 12 reference words.

Model	En-De SMT					En-De NMT				
	MT	gaps	source	r	ρ	MT	gaps	source	r	ρ
QUETCH	39.90	17.10	36.10	48.32	51.31	29.18	13.26	28.91	42.84	49.59
NUQE	50.04	35.53	42.08	59.62	60.89	32.49	15.01	30.19	43.41	50.87
APE-QE	55.12	47.04	51.11	58.01	60.58	37.60	21.78	34.46	35.23	38.88
PRED-EST	57.29	43.68	33.02	70.95	74.49	39.25	21.54	29.52	50.18	55.66
ENSEMBLED	61.33	53.05	51.11	72.89	76.37	43.04	24.74	34.46	52.34	56.98
STACKED	62.40	-	-	-	-	43.88	-	-	-	-

Table 1: Benchmarking of the different models implemented in OpenKiwi on the WMT 2018 development set, along with an ensembled system that averages the outputs of the other systems (ENSEMBLED) and a stacked architecture with stacks their predictions into a linear feature-based model (STACKED). For each system, we report the five official scores used in WMT 2018: word-level F_1^{mult} for MT, gaps, and source tokens, and sentence-level Pearson’s r and Spearman’s ρ rank correlations.

shared task data only (i.e., no auxiliary parallel or roundtrip data).

APE-QE. Automatic post-editing (APE) has been used by [Martins et al. \(2017\)](#) as an intermediate step for quality estimation, where an APE system is trained on the human post-edits and its outputs are used as pseudo-post-editions to generate word-level quality labels and sentence-level scores in the same way that the original labels were created. Since OpenKiwi’s focus is not on implementing a sequence-to-sequence model, we used an external software, OpenNMT-py ([Klein et al., 2017](#)), to train two separate translation models:

- SRC \rightarrow PE: trained first on the in-domain corpus provided, then fine-tuned on the shared task data.
- MT \rightarrow PE: trained on the concatenation of

the corpus of artificially created sentence triplets and the shared task data oversampled by a factor of 20.

These predictions are then combined in the ensemble and stacked systems as explained below.

Predictor-Estimator. Our implementation follows closely the architecture proposed by [Kim et al. \(2017\)](#), which consists of two modules:

- a *predictor*, which is trained to predict each token of the target sentence given the source and the left and right context of the target sentence;
- an *estimator*, which takes features produced by the *predictor* and uses them to classify each word as OK or BAD.

Model	En-De SMT					En-De NMT				
	MT	gaps	source	r	ρ	MT	gaps	source	r	ρ
deepQUEST	42.98	28.24	33.97	48.72	50.97	30.31	11.93	28.59	38.08	48.00
UNQE	–	–	–	70.00	72.44	–	–	–	51.29	60.52
Wang et al. (2018)	62.46	49.99	–	73.97	75.43	43.61	–	–	50.12	60.49
OpenKiwi	62.70	52.14	48.88	71.08	72.70	44.77	22.89	36.53	46.72	58.51

Table 2: Final results on the WMT 2018 test set. deepQUEST is the open-source system developed by [Ive et al. \(2018\)](#), and UNQE is the unpublished system from Jiangxi Normal University, described by [Specia et al. \(2018a\)](#).

Our predictor uses a bidirectional LSTM to encode the source, and two unidirectional LSTMs processing the target in left-to-right (LSTM-L2R) and right-to-left (LSTM-R2L) order. For each target token t_i , the representations of its left and right context are concatenated and used as query to an attention module before a final softmax layer. It is trained on the large parallel corpora provided as additional data by the WMT shared task organizers. The estimator takes as input a sequence of features: for each target token t_i , the final layer before the softmax (before processing t_i), and the concatenation of the i -th hidden state of LSTM-L2R and LSTM-R2L (after processing t_i). In addition, we train this system with a multi-task architecture that allows us to predict sentence-level HTER scores. Overall, this system is capable to predict sentence-level scores and all word-level labels (for MT words, gaps, and source words)—the source word labels are produced by training a predictor in the reverse direction.

Stacked Ensemble. Finally, we ensemble the systems above using a stacked architecture with a feature-based linear system, as described by [Martins et al. \(2017\)](#). The features are the ones described there, including lexical and part-of-speech tags from words, their contexts, and their aligned words and contexts, as well as syntactic features and features provided by a language model (as provided by the organizers). This system is only used to produce word-level labels for MT words.

5 Benchmark Experiments

We show benchmark numbers on the two English-German WMT 2018 datasets. In Table 1, we compare different configurations of OpenKiwi on the development datasets. For the single systems, we can see that the predictor-estimator has the best performance, except for predicting the source and

the gap word-level tags, where APE-QE is superior. Overall, ensembled versions of these systems perform the best, with a stacked architecture being very effective for predicting word-level MT labels, confirming the findings of [Martins et al. \(2017\)](#).

Finally, in Table 2, we report numbers on the official test set. We compare OpenKiwi against the best systems in WMT 2018 ([Specia et al., 2018a](#)) and another existing open-source tool, deepQuest ([Ive et al., 2018](#)). Overall, OpenKiwi outperforms deepQuest for all word-level and sentence-level tasks, and attains the best results for all the word-level tasks.

6 Conclusions

We presented OpenKiwi, a new open-source framework for QE. OpenKiwi is implemented in Pytorch and it supports training of word-level and sentence-level QE systems on new data. It outperforms other open-source toolkits on both word-level and sentence-level, and it yields state-of-the-art word-level QE results.

Acknowledgments

The authors would like to thank Eduardo Fierro, Thomas Reynaud, and the Unbabel AI and Engineering teams for their invaluable contributions to OpenKiwi.

References

- Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.
- Julia Ive, Frédéric Blain, and Lucia Specia. 2018. DeepQuest: a framework for neural-based Quality Estimation. In *International Conference on Computational Linguistics (COLING)*.

- Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-Estimator using Multilevel Task Learning with Stack Propagation for Neural Quality Estimation. In *Conference on Machine Translation (WMT)*.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.
- Julia Kreutzer, Shigehiko Schamoni, and Stefan Riezler. 2015. QQuality Estimation from ScraTCH (QUETCH): Deep Learning for Word-level Translation Quality Estimation. In *Workshop on Machine Translation (WMT)*.
- Varvara Logacheva, Chris Hokamp, and Lucia Specia. 2016. Marmot: A toolkit for translation quality estimation at the word level. In *LREC*.
- André F. T. Martins, Ramon Astudillo, Chris Hokamp, and Fábio Kepler. 2016. Unbabel’s Participation in the WMT16 Word-Level Translation Quality Estimation Shared Task. In *Conference on Machine Translation (WMT)*.
- André F. T. Martins, Marcin Junczys-Dowmunt, Fabio Kepler, Ramon Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017. Pushing the limits of translation quality estimation. *Transactions of the Association for Computational Linguistics (to appear)*.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.
- Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón Astudillo, and André F. T. Martins. 2018a. [Findings of the wmt 2018 shared task on quality estimation](#). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 702–722, Belgium, Brussels. Association for Computational Linguistics.
- Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. [Multi-level translation quality prediction with quest++](#). In *ACL-IJCNLP 2015 System Demonstrations*, pages 115–120, Beijing, China.
- Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018b. Quality estimation for machine translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.
- Jiayi Wang, Kai Fan, Bo Li, Fengming Zhou, Boxing Chen, Yangbin Shi, and Luo Si. 2018. Alibaba Submission for WMT18 Quality Estimation Task. In *Conference on Machine Translation (WMT)*.