deepQuest-py: Large and Distilled Models for Quality Estimation

Fernando Alva-Manchego,[1] Abiola Obamuyide,[1] Amit Gajbhiye,[1] Frédéric Blain,[1,2]
Marina Fomicheva,[1] Lucia Specia[1,3]
[1]University of Sheffield, [2]University of Wolverhampton, [3]Imperial College London
{f.alva,a.obamuyide,a.gajbhiye,m.fomicheva}@sheffield.ac.uk
f.blain@wlv.ac.uk, l.specia@imperial.ac.uk

Abstract

We introduce deepQuest-py, a framework for training and evaluation of large and light-weight models for Quality Estimation (QE). deepQuest-py provides access to (1) state-of-the-art models based on pre-trained Transformers for sentence-level and word-level QE; (2) light-weight and efficient sentence-level models implemented via knowledge distillation; and (3) a web interface for testing models and visualising their predictions. deepQuest-py is available at https://github.com/sheffieldnlp/deepQuest-py under a CC BY-NC-SA licence.

1 Introduction

Quality Estimation (QE) for Machine Translation (MT) aims to predict how good automatic translations are without comparing them to gold-standard references (Specia et al., 2009). This is useful in real-world scenarios (e.g. computer-aided translation or online translation of social media content), where users would benefit from knowing how confident they should be of the generated translations. QE has received increased attention in the MT community, with Shared Tasks being organised yearly since 2012 as part of WMT, the main conference in MT research (Callison-Burch et al., 2012; Bojar et al., 2013, 2014, 2015, 2016, 2017; Specia et al., 2018a; Fonseca et al., 2019; Specia et al., 2020).

Given an original-translation sentence pair, QE scores can be computed in different granularities (Specia et al., 2018b). At the **word-level**, each word in the original and/or translation sentence receives a tag indicating whether it was correctly translated or not (e.g. OK or BAD). Gaps in the translation could also receive labels to indicate when a word is missing. At the **sentence-level**, a single continuous score is predicted for each original-translation pair. For example, 0-100 for direct assessments (DA, Graham et al., 2017), or 0-1 for human-targeted translation error rate (HTER, Snover et al., 2006).

Few open-source software is available for implementing QE models. QuEst (Specia et al., 2013) and QuEst++ (Specia et al., 2015) were the first ones, and included methods that relied on extracting linguistically-motivated features to train traditional machine learning models (e.g. support vector machines). With the advent of neural-based approaches, deepQuest (Ive et al., 2018) provided a TensorFlow-based framework for RNN-based sentence-level and document-level QE models, inspired by the Predictor-Estimator approach (Kim et al., 2017). OpenKiwi (Kepler et al., 2019) implements a common API for experimenting with several feature-based and neural-based QE models. More recently, TransQuest (Ranasinghe et al., 2020b) released state-of-the-art models for sentence-level QE based on pre-trained Transformer architectures.

As shown in the latest WMT20 QE Shared Task (Specia et al., 2020), systems are increasingly relying on large pre-trained models to achieve impressive results in the different proposed tasks. However, their considerable size could prevent their application in scenarios where fast inference is required and small disk space is available. To overcome this limitation, Gajbhiye et al. (2021) propose to use Knowledge Distillation (KD, Hinton et al., 2015) to transfer knowledge from a large top-performing *teacher* model into a smaller (in terms of memory print, computational power and prediction latency) yet well-performing *student* model. The authors applied this framework to QE, and effectively trained light-weight QE models with similar performance to SoTA architectures trained on distilled yet large pre-trained representations.

In this paper, we introduce deepQuest-py, a new version of deepQuest that covers both large and light-weight neural QE models, with a particular emphasis on knowledge distillation. The main fea-

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 382–389
November 7–11, 2021. ©2021 Association for Computational Linguistics

tures of deepQuest-py are:

- Implementation of state-of-the-art models for sentence-level (Ranasinghe et al., 2020a) and word-level (Lee, 2020) QE;
- The first implementation of light-weight sentence-level QE models using knowledge distillation (Gajbhiye et al., 2021);
- Easy-to-use command-line interface and API to train and test QE models in custom datasets, as well as those from several WMT QE Shared Tasks thanks to its integration with HuggingFace Datasets (Lhoest et al., 2021); and
- An online tool to try out trained models, evaluate them and visualise their predictions.

Different from existing open-source toolkits in the area, our aim is to provide access to neural QE models for both researchers (via a command-line interface and python library) and end-users (via a web-based tool). Additionally, this is the only tool to provide implementation of knowledge distillation for QE. In the following sections, we detail the main functionalities offered by deepQuest-py: implementation of state-of-the-art sentence-level and word-level models (Sec. 2); implementation of light-weight sentence-level models through knowledge distillation (Sec. 3); and evaluation and visualisation of models' predictions via a web interface (Sec. 4). We expect that deepQuest-py facilitates the implementation of QE models, allows useful analysis of their capabilities, and promotes their adoption by end-users.

2 Large State-of-the-Art Models

In the WMT20 QE Shared Task (Specia et al., 2020), the top performing models were based on fine-tuning pre-trained Transformers (Vaswani et al., 2017), such as BERT (Devlin et al., 2019) or XLM-R (Conneau et al., 2020). deepQuest-py provides access to this type of approaches by building on the HuggingFace Transformers (Wolf et al., 2020)[1] library. We provide implementations for sentence-level and word-level QE.

Sentence-Level. deepQuest-py implements the MonoTransQuest architecture from TransQuest (Ranasinghe et al., 2020a,b), the overall winner in Task 1 (sentence-level direct assessment) of the WMT20 QE Shared Task. In this approach, the original sentence and its translation are concatenated using the [SEP] token, and passed through XLM-R to obtain a joint representation via the [CLS] token. This serves as input to a softmax layer that is used to predict translation quality. In order to boost performance, the authors incorporate two strategies: (1) to use an ensemble of two models: one that fine-tunes XLM-R-base and one that fine-tunes XLM-R-large; and (2) to augment the training data of the QE models with (subsets of) the training data of the NMT models, considering their quality scores as perfect. These extensions are not currently available in deepQuest-py, but the API is flexible-enough to incorporate them in the future.

Word-Level. deepQuest-py implements the model proposed by BeringLab (Lee, 2020), the winner of Task 2 Word-level QE for En-De in the WMT20 QE Shared Task. Similar to the sentence-level models described before, the original sentence and its translation are fed to XLM-R to get contextualised word embeddings. In this approach, both token-level (hidden states) and instance-level ([CLS] token) representations are used as input to dedicated linear layers that predict word-level and sentence-level quality estimates, respectively. The model is trained jointly in these two tasks. In order to boost performance, this approach creates artificial data in a similar fashion to Negri et al. (2018). Given a dataset of parallel source-target sentences, an NMT model is trained using 90% of the data. Then, the NMT model translates source sentences in the remaining 10% of the data. After that, HTER word labels are generated for this 10% of the data, leveraging their manual references as if they were post-edits of the translations generated by the NMT model.deepQuest-py includes scripts and examples for executing this training pipeline, provided that the user has access to automatic translations. For obtaining word-level tags, in particular, deepQuest-py leverages publicly-available scripts.[2]

3 Light-Weight Distilled Models

The most distinctive contribution in deepQuest-py is the **ability to train light-weight and efficient QE models through knowledge distillation**. We

[1] https://github.com/huggingface/transformers

[2] https://github.com/deep-spin/qe-corpus-builder

Name	Training data	Et-En	Ro-En	Si-En	Ne-En	En-Zh
TQ$_{\text{TEACHER}}$	MLQE-gold	0.77	0.88	0.60	0.75	0.44
BiRNN$_{\text{STUDENT}}$	MLQE-dist	0.45	0.62	0.44	0.46	**0.18**
BiRNN$_{\text{STUDENT+AUG}}$	Wiki-dist	**0.50**	**0.69**	**0.45**	**0.54**	0.17
BiRNN	MLQE-gold	0.37	0.60	0.40	0.42	0.15
Predictor-Estimator	MLQE-gold	0.48	0.69	0.37	0.39	0.19
TQ$_{\text{DistilBERT}}$	MLQE-gold	0.62	0.78	0.51	0.61	0.36

Table 1: Pearson correlation with human judgments on the MLQE test set. MLQE-gold: training partition of MLQE dataset; MLQE-dist: distilled version of the MLQE training set with teacher predictions used as labels; Wiki-dist: the Wikipedia dataset produced by data augmentation. Boldface results indicate our best student models.

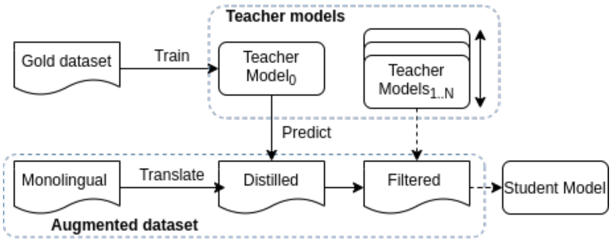


Figure 1: Knowledge distillation with data augmentation and noise filtering based on teacher uncertainty (Gajbhiye et al., 2021).

Name	#params	Inference		
		Speed (secs.)	RAM (MiB)	Disk (M)
TQ$_{\text{XLM-R-Large}}$	561M	0.82	9,263.5	2140
TQ$_{\text{DistilBERT}}$	135M	1.09	1,979.2	517
BiRNN	18M	0.39	155.6	132

Table 2: Efficiency. Inference speed and RAM for prediction are for 1 sentence on CPU (Intel Xeon Silver 4114 CPU @ 2.20GHz).

implement the approach proposed by Gajbhiye et al. (2021) to directly distil **sentence-level** QE models, where the student architecture can be completely different from that of the teacher. Namely, they distill large and powerful QE models based on XLM-R into small bidirectional RNN-based models.

3.1 BiRNN-based Architecture

deepQuest-py implements sentence-level models following the architecture proposed by Ive et al. (2018). In this approach, the original sentence and its translation are encoded independently using dedicated BiRNNs. To obtain predictions, these two representations are concatenated as the weighted sum of their word vectors, generated by an attention mechanism. Then, this joint representation is passed through a dense layer with sigmoid activation to generate the quality estimates. deepQuest-py uses AllenNLP (Gardner et al., 2018)[3] as its backbone for the BiRNN model.

3.2 Knowledge Distillation

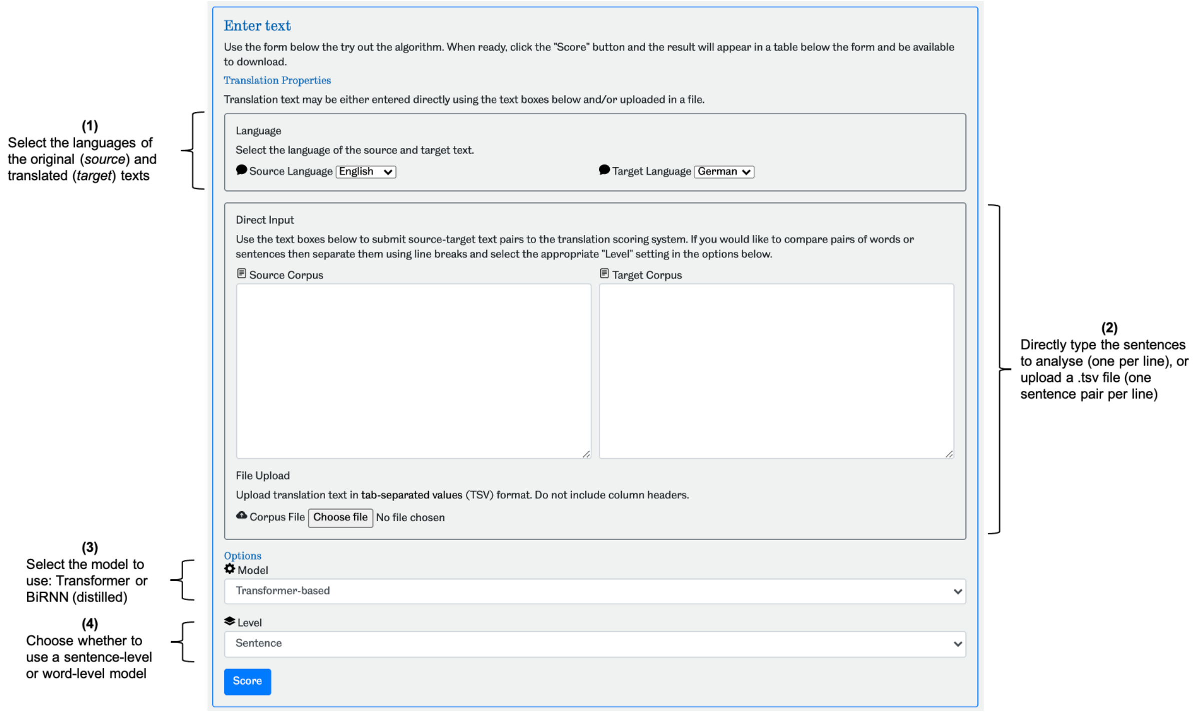
For cases where large size SotA QE models are not deployable, Gajbhiye et al. (2021) propose to use a KD approach to train more efficient and well-performing models for sentence-level QE. The approach (illustrated in Figure 1) consists of three steps described below.

Teacher-Student Training. A large SotA QE model generates predictions on a gold training dataset, and these are directly used to train a light-weight model. Gajbhiye et al. (2021) employs pre-trained Transformer models (such as those from Sec. 2) as teachers, and BiRNN models (such as those from Sec. 3.1) as students. Table 2 shows the number of parameters, memory and disk space requirements, as well as inference speed for the teacher model (TQ$_{\text{XLM-R-Large}}$), student model (BiRNN) and a MonoTransQuest model built on DistilBERT (TQ$_{\text{DistilBERT}}$), using data in the MLQE Et-En dataset.

Data Augmentation. Teacher predictions on the gold training dataset may prove insufficient to train the student. Therefore, Gajbhiye et al. (2021) collect additional monolingual data and: (1) translate it with the same MT models used for the gold data, and (2) generate QE scores with the teacher model.

Noise Filtering. Teacher predictions can be noisy and degrade student performance. To overcome this, Gajbhiye et al. (2021) propose a filtering approach based on uncertainty quantification over predictions from an ensemble of teacher models. Concretely, they propose to: (1) train several teacher models on the same dataset with different random initialisations; (2) generate several teacher

[3] `https://allennlp.org/`



Enter text

Use the form below the try out the algorithm. When ready, click the "Score" button and the result will appear in a table below the form and be available to download.

Translation Properties

Translation text may be either entered directly using the text boxes below and/or uploaded in a file.

Language

Select the language of the source and target text.

Source Language: English

Target Language: German

Direct Input

Use the text boxes below to submit source-target text pairs to the translation scoring system. If you would like to compare pairs of words or sentences then separate them using line breaks and select the appropriate "Level" setting in the options below.

File Upload

Upload translation text in tab-separated values (TSV) format. Do not include column headers.

Corpus File: No file chosen

Options

Model: Transformer-based

Level: Sentence

(1) Select the languages of the original (*source*) and translated (*target*) texts

(2) Directly type the sentences to analyse (one per line), or upload a .tsv file (one sentence pair per line)

(3) Select the model to use: Transformer or BiRNN (distilled)

(4) Choose whether to use a sentence-level or word-level model

Figure 2: Submission form for the web tool.

predictions for each instance in the student training data; and (3) filter out instances for which the predictions show high variance (i.e. the variance is more than one standard deviation away from its mean).

On experiments performed using the MLQE dataset (Fomicheva et al., 2020), Gajbhiye et al. (2021) show that their approach results in QE models that are 4x smaller in disk space with 8x fewer parameters, and 3x faster in inference speed than large SotA Transformer-based models. In particular, as shown in Table 1, distilled models with augmented data achieved comparable performances to training a large model on DistilBERT ($\text{TQ}_{\text{DistilBERT}}$), but with a lighter BiRNN-based architecture. In addition, this approach allows for substantial improvements over shallow models trained on gold data (BiRNN and Predictor-Estimator) for all of the language pairs. For further details, we refer the reader to (Gajbhiye et al., 2021).

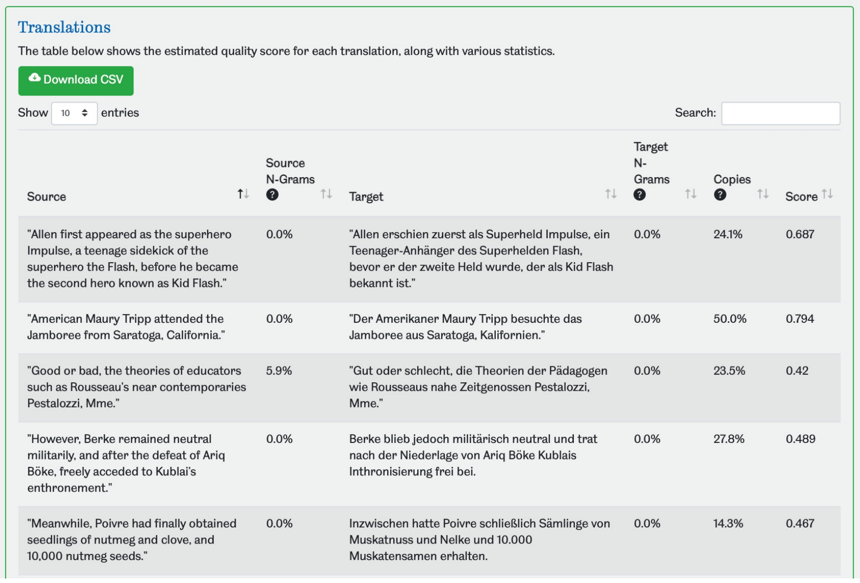
deepQuest-py provides command-line functionalities for all steps in the KD pipeline.

4 Web Tool for Analysis and Visualisation

deepQuest-py offers a demo web-service with visualization for sentence-level and word-level QE models. The user fills in a simple submission form (Figure 2) indicating: the languages of the original sentences and their translations, the sentences to analyse (either typing them directly or through a .tsv file), the type of model to use for prediction (e.g. Transformer or BiRNN), and the level of granularity for the scores (sentence-level or word-level). After pressing the *score* button, the user is presented with varied results for their analysis:

Scores per Instance. For sentence-level predictions, each submitted original-translation pair is shown alongside its estimated quality score (Figure 3). The value for this score and its interpretation depends on the data that the QE model was trained on. For example, models trained on HTER scores will output values between 0 and 1, with lower scores indicating better quality. On the other hand, models trained on normalised DA scores will output negative and positive values – the higher the better.[4] Two additional metrics are included: the proportion of repeated n-grams in the source/target (named source/target n-grams), and the proportion of words in the target sentence that are copies of words in the source (i.e. untranslated words). The user can navigate through the analysed sentences

[4] While raw DA scores have a 0-100 value range, their normalised values will depend on the actual minimum and maximum scores in the training data.



Translations

The table below shows the estimated quality score for each translation, along with various statistics.

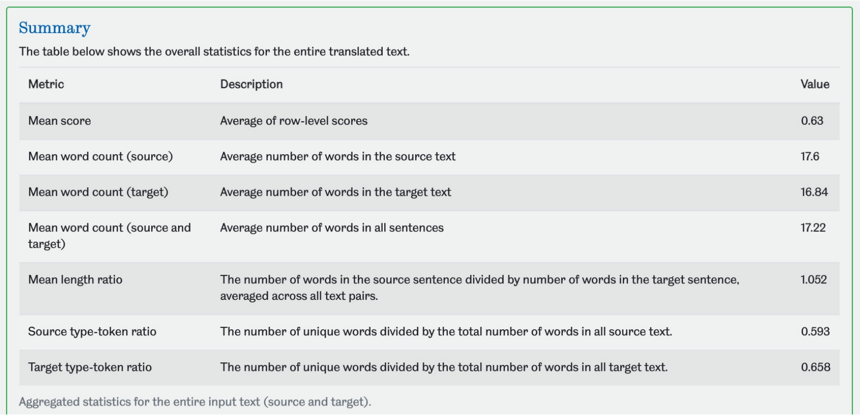
Download CSV

Show 10 entries

Search:

Source	Source N-Grams	Target	Target N-Grams	Copies	Score
"Allen first appeared as the superhero Impulse, a teenage sidekick of the superhero the Flash, before he became the second hero known as Kid Flash."	0.0%	"Allen erschien zuerst als Superheld Impulse, ein Teenager-Anhänger des Superhelden Flash, bevor er der zweite Held wurde, der als Kid Flash bekannt ist."	0.0%	24.1%	0.687
"American Maury Tripp attended the Jamboree from Saratoga, California."	0.0%	"Der Amerikaner Maury Tripp besuchte das Jamboree aus Saratoga, Kalifornien."	0.0%	50.0%	0.794
"Good or bad, the theories of educators such as Rousseau's near contemporaries Pestalozzi, Mme."	5.9%	"Gut oder schlecht, die Theorien der Pädagogen wie Rousseau nahe Zeitgenossen Pestalozzi, Mme."	0.0%	23.5%	0.42
"However, Berke remained neutral militarily, and after the defeat of Ariq Böke, freely acceded to Kublai's enthronement."	0.0%	Berke blieb jedoch militärisch neutral und trat nach der Niederlage von Ariq Böke Kublais Inthronisierung frei bei.	0.0%	27.8%	0.489
"Meanwhile, Poivre had finally obtained seedlings of nutmeg and clove, and 10,000 nutmeg seeds."	0.0%	Inzwischen hatte Poivre schließlich Sämlinge von Muskatnuss und Nelke und 10.000 Muskatensamen erhalten.	0.0%	14.3%	0.467

Figure 3: Scores per instance.



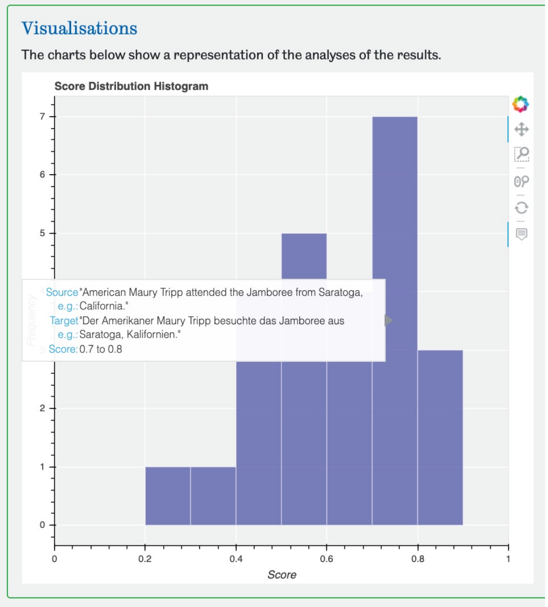
Summary

The table below shows the overall statistics for the entire translated text.

Metric	Description	Value
Mean score	Average of row-level scores	0.63
Mean word count (source)	Average number of words in the source text	17.6
Mean word count (target)	Average number of words in the target text	16.84
Mean word count (source and target)	Average number of words in all sentences	17.22
Mean length ratio	The number of words in the source sentence divided by number of words in the target sentence, averaged across all text pairs.	1.052
Source type-token ratio	The number of unique words divided by the total number of words in all source text.	0.593
Target type-token ratio	The number of unique words divided by the total number of words in all target text.	0.658

Aggregated statistics for the entire input text (source and target).

Figure 4: Statistics summarising the scores for all submitted sentences.

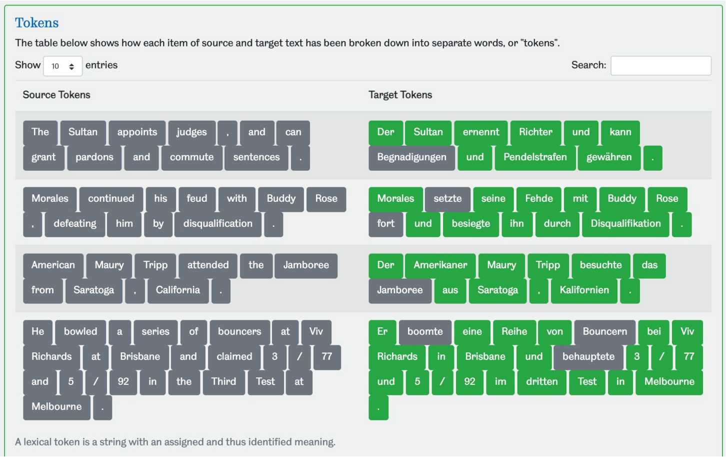


Visualisations

The charts below show a representation of the analyses of the results.

Source "American Maury Tripp attended the Jamboree from Saratoga,
e.g.: California."
Target "Der Amerikaner Maury Tripp besuchte das Jamboree aus
e.g.: Saratoga, Kalifornien."
Score: 0.7 to 0.8

Figure 5: Histogram with the distribution of predicted scores in the dataset.



Tokens

The table below shows how each item of source and target text has been broken down into separate words, or "tokens".

Show 10 entries

Search:

Source Tokens	Target Tokens
The Sultan appoints judges , and can grant pardons and commute sentences .	Der Sultan ernennt Richter und kann Begnadigungen und Pendelstrafen gewähren .
Morales continued his feud with Buddy Rose , defeating him by disqualification .	Morales setzte seine Fehde mit Buddy Rose fort und besiegte ihn durch Disqualifikation .
American Maury Tripp attended the Jamboree from Saratoga , California .	Der Amerikaner Maury Tripp besuchte das Jamboree aus Saratoga , Kalifornien .
He bowled a series of bouncers at Viv Richards at Brisbane and claimed 3 / 77 and 5 / 92 in the Third Test at Melbourne .	Er boomte eine Reihe von Bouncern bei Viv Richards in Brisbane und behauptete 3 / 77 und 5 / 92 im dritten Test in Melbourne .

A lexical token is a string with an assigned and thus identified meaning.

Figure 6: Visualisation of tokens in original (source) and translation (target) sentences. For the translation side, tokens in green are predicted to have an 'OK' label, while tokens in grey, a 'BAD' label. In this example, the model did not produce word-level predictions for the original sentences.

using the form shown (including searching for instances with specific words), or download all the information to a .csv file.

Scores Summary. A table summarises the scores for all the submitted sentences providing some simple statistics. See Figure 4 for the names and descriptions of the metrics considered.

Scores Distribution. The web tool also shows a histogram with the distribution of the sentence-level scores over all submitted instances (Figure 5). The user can hover over the bars in the plot to see examples of original-translation pairs whose scores correspond to the selected range.

Token Annotations. The web tool shows the tokenisation of each original-translation pair (Figure 6). For word-level predictions, in particular, the predicted quality label for each token is shown in different colours: green for 'OK' and grey for 'BAD'. The user can also search within the instances looking for sentences with specific words.

The demo web tool includes Transformer-based sentence-level models for all language pairs in the MLQE dataset: English-German, English-Chinese, Romanian-English, Estonian-English, Nepalese-English, Sinhala-English, and Russian-English. There is also the option to use a multilingual model. For Transformer-based word-level predictions, only an English-German model is available. BiRNN-based (distilled) models for sentence-level QE are available for a subset of languages. We note that our purpose with this paper is not to provide prediction models for multiple languages, but rather to demonstrate the functionalities of the back- and front-end in deepQuest-py.

The demo showcasing all the functionalities offered by deepQuest-py is available at `https://github.com/sheffieldnlp/deepQuest-py/tree/main/demo`.

5 Conclusions

We have presented deepQuest-py, a new framework for implementation and evaluation of QE models. On top of large state-of-art models based on pre-trained Transformer architectures, deepQuest-py targets the development of light-weight and efficient models around the teacher-student framework for knowledge distillation. In addition, deepQuest-py encourages end-user adoption of QE technologies by providing a web application to obtain quality predictions and analyse model performance.

Acknowledgements

This work was supported by funding from the Bergamot project (EU H2020 Grant No. 825303).

References

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Erick Fonseca, Lisa Yankovskaya, André 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 1–10, Florence, Italy. Association for Computational Linguistics.

Amit Gajbhiye, Marina Fomicheva, Fernando Alva-Manchego, Frédéric Blain, Abiola Obamuyide, Nikolaos Aletras, and Lucia Specia. 2021. Knowledge distillation for quality estimation. In *Findings of the Association for Computational Linguistics: ACL 2021*. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, 23(1):3–30.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Julia Ive, Frédéric Blain, and Lucia Specia. 2018. deepQuest: A framework for neural-based quality estimation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3146–3157, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Fabio Kepler, 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. Association for Computational Linguistics.

Hyun Kim, 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. Association for Computational Linguistics.

Dongjun Lee. 2020. Two-phase cross-lingual language model fine-tuning for machine translation quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1024–1028, Online. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Patrick von Platen, Thomas Wolf, Yacine Jernite, Abhishek Thakur, Lewis Tunstall, Suraj Patil, Mariama Drame, Julien Chaumond, Julien Plu, Joe Davison, Simon Brandeis, Teven Le Scao, Victor Sanh, Kevin Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Steven Liu, Sylvain Lesage, Lysandre Debut, Théo Matussière, Clément Delangue, and Stas Bekman. 2021. huggingface/datasets: 1.11.0.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. ESCAPE: a large-scale synthetic corpus for automatic post-editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020a. TransQuest at WMT2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055, Online. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020b. TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón F. 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: Shared Task Papers*, pages 689–709, Belgium, Brussels. Association for Computational Linguistics.

Lucia Specia, Gustavo 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. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018b. Quality estimation for machine translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.

Lucia Specia, Kashif Shah, Jose G.C. de Souza, and Trevor Cohn. 2013. QuEst - a translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84, Sofia, Bulgaria. Association for Computational Linguistics.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.