# An efficient and user-friendly tool for machine translation quality estimation

**Kashif Shah**§, **Marco Turchi**†, **Lucia Specia**§

§Department of Computer Science, University of Sheffield, UK
{kashif.shah,l.specia}@sheffield.ac.uk
†Fondazione Bruno Kessler, University of Trento, Italy
turchi@fbk.eu

## Abstract

We present a new version of QuEst – an open source framework for machine translation quality estimation – which brings a number of improvements: (i) it provides a Web interface and functionalities such that non-expert users, e.g. translators or lay-users of machine translations, can get quality predictions (or internal features of the framework) for translations without having to install the toolkit, obtain resources or build prediction models; (ii) it significantly improves over the previous runtime performance by keeping resources (such as language models) in memory; (iii) it provides an option for users to submit the source text only and automatically obtain translations from Bing Translator; (iv) it provides a ranking of multiple translations submitted by users for each source text according to their estimated quality. We exemplify the use of this new version through some experiments with the framework.

**Keywords:** Machine Translation, Translation Evaluation, Translation Quality Estimation

## 1. Introduction

Metrics to predict the quality of texts translated automatically by Machine Translation (MT) systems have become a necessity in many scenarios. These metrics, referred to as quality estimation (QE), or also *confidence estimation*, are aimed at MT systems in use. They consist in prediction models generally built using supervised machine learning algorithms from examples of source texts and their machine translations (i.e., no access to reference translations) described through a number of features and labelled for quality. The notion of "quality" in QE metrics is defined according to the application and represented by labels – post-editing effort, gisting reliability, etc. – and features – for example, a binary grammar checker feature will be important for fluency prediction, but less useful for gisting reliability prediction.

A number of positive results have been reported in recent work in the field. Examples include improving post-editing efficiency by filtering out low quality segments which would require more effort or time to correct than translating from scratch (Specia et al., 2009; Specia, 2011), selecting high quality segments to be published as they are, without post-editing (Soricut and Echihabi, 2010), selecting a translation from either an MT system or a translation memory for post-editing (He et al., 2010), selecting the best translation from multiple MT systems (Specia et al., 2010; Avramidis, 2013), and highlighting sub-segments that need revision (Bach et al., 2011). For recent overviews of various algorithms and features we refer the reader to the WMT12-13 editions of the shared task on QE (Callison-Burch et al., 2012; Bojar et al., 2013).

QuEst (Specia et al., 2013) is an open-source framework for QE which provides a wide range of feature extractors from source and translation texts, as well as external resources and tools. These lead to an average of 150 features (depending on the language pair) and go from simple, language-independent features, to advanced, linguistically motivated features. They include features that rely on information from the MT system that generated the translations, and features that are oblivious to the way translations were produced. In addition, QuEst integrates a well-known machine learning toolkit, `scikit-learn`,[1] and other algorithms that are known to perform well on this task, facilitating experiments with existing features and techniques for feature selection and model building. QuEst also provides documentation for users to add their own features and learning algorithms. However, QuEst is not directly usable by end-users, such as professional translators. The tasks of installing and configuring the toolkit, obtaining the necessary resources, and building new models from data require technical knowledge of natural language processing, machine translation and machine learning.

In this paper we describe a number of improvements over the current version of QuEst which are meant to make it more accessible to non-expert users, as well as more efficient (i.e., faster). In particular, we provide

[1] `http://scikit-learn.org/`

a client-server architecture which allows users to access pre-built models and resources remotely through XML-RPC requests, and which is optimised for speed by keeping resources in memory (Section 2.); a Web interface where users can upload files with source and translation segments, with the possibility of getting translations from Bing Translator (Section 3.); and a ranking mechanism that provides a sorted list of multiple the options of translations given for each source segment based on their predicted quality (Section 4.).

## 2. Client-server architecture

The adaptation of QuEst to the online scenario has required an upgrade of different components in the previous version. The main goals of such changes were to: i) allow the processing of one sentence pair at the time; ii) speed up the feature extraction; iii) reduce the usage of computational resources; iv) make QuEst easily accessible remotely.

**Code Refactoring** The previous version of QuEst was designed to process text files containing multiple source and target sentences. This required the following steps to be performed at run time, before any feature could start being extracted:

1. Loading in memory of the main resources needed to extract features, such as a list of the n-grams from the MT training corpus, source and target language models, and bilingual dictionaries.
2. Pre-processing of the whole source and target files extracting information such as part-of-speech tagging and language model probabilities.
3. Filtering of the main resources according to the source and target sentences to reduce the computational effort while extracting features.

Only when these steps were completed for the entire input files, the extraction of features for each sentence pair could start. In the current version of QuEst, this structure has been refactored for efficiency when processing a large number of sentences, and to better fit the demand of the interface: processing one sentence pair at the time. Changes were necessary in the first two steps, with the last step being removed. Dealing with on the fly requests of predictions for a given sentence pair does not allow the pre-filtering of resources. This modification has made the resources stored in memory completely independent of the sentences to be processed. The loading in memory of the resources is now part of an initialisation step. This modification has increased the amount of memory required to store all resources, but on the other hand it made QuEst more suitable to be embedded in a client-server framework, and to deal with any unseen sentence pair.

**Language Model Servers** Some of the most effective features for QE require the computation of sentence level language model (LM) probabilities and perplexities. In general, effective LMs are obtained from large corpora, and can thus can be very large files as well. This implies that starting the LMs during the QuEst computation and having them running on the same machine can be problematic. To cope with these problems, multiple LMs can be distributed in various servers which can run independently from QuEst, in different machines. Within QuEst, the computation of LM scores is done in a client that, given a sentence, queries the LM server to get the relevant scores. In the initialisation step, the connections with the LM servers are established and a fake query is used to force the initialisation of the LMs.

**Client-Server Framework** To conclude the adaptation, the new version of QuEst has been embedded in a server that allows connection to the feature extractor from different clients located in various machines. This wrapper links QuEst to external machines using sockets, while it is linked to QuEst using standard input and output streams. The new structure of QuEst is outlined in Figure 1.

**Offline Pre-processing** Launching external software within QuEst, such as tokenization and true-casing, is time consuming. To mitigate this effect, QuEst can now easily deal with already pre-processed source and target sentences.

These modifications have made QuEst slimmer and easier to be used. In particular, they have speeded up the feature extraction process allowing its use in an online scenario or as a part of a Web interface, as we discuss in the next Section.

## 3. An interface to QuEst

In order to facilitate the use of QuEst by non-expert end-users, such as translators or users of online MT systems, we have developed a Web interface that allows users to access the tool remotely from a Web browser, without the need to understand the internal functioning of framework, nor to install/configure the tool or build models. It offers the following functionalities:

- **Features**: Values for individual features describing the source and translation sentence, e.g. source and target length, LM scores, average translation ambiguity level of source words, etc.

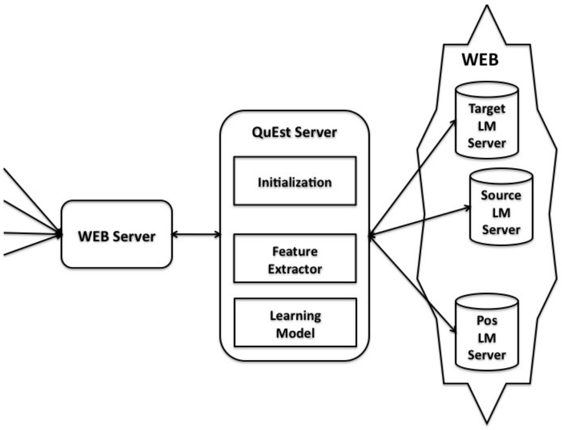

Figure 1: Client-Server skema.

- **Predictions**: An estimated quality score for the translated sentence given the source sentence, produced using Support Vector Regression (SVR) and pre-built models for specific language pairs.
- **Ranking**: The ranking of multiple translations submitted by the user for a given source sentence. This is done based on SVR quality predictions for each source-target sentence pair performed independently.

The Web interface is developed using PHP and XML-RPC for communication across the main QUEST server and resource servers. For the convenience of users, we have also integrated the free Bing translation API to this Web interface.[2]

The pipeline of the framework accessed via its Web interface is the following:

- User inputs a file containing source sentences only or tab separated source sentences and their translation(s) – as many translations as desired.
- User selects the language pair and text type (domain, etc.).
- File is uploaded to the Web server, and read line by line (sentence by sentence).
- If the input file contains only source sentences, a request is sent to Bing's API with the selected target language.
- Based on the choices (language pair, text type) selected by the user, an instance of QUEST with the appropriate prediction model and resources is triggered.
- QUEST extracts the features by calling the Feature Extractor module. LMs, other resources and prediction models are already loaded into memory by a fake call.
- QUEST generates a prediction for each source-target combination by applying the prediction model for that language pair.
- If the input file contains multiple translations for the same source sentence, QUEST ranks these translations.

These functionalities require prediction models previously trained offline for each language pair of interest. Options to build models from examples of translations, quality scores and language resources will be added to the interface in the future.

## 4. Experiments

QUEST has recently been benchmarked on a number of datasets (Shah et al., 2013b; Shah et al., 2013a). To make this paper self-contained, we provide experiments with models trained offline which are already available through the Web interface. We also present figures in terms of the running time of our models.

Our experiments include two language pairs, i.e., French-English and German-English, and two different tasks: absolute quality scores prediction and ranking of up to five translation options. The data used and results for these tasks are given in Tables 1 and 2. For both tasks, a set of 17 well established *baseline* features was used.[3] The language models and other resources were built using standard tools: SRILM (Stolcke, 2002) and GIZA++ (Och and Ney, 2003). SVR

[2] Please note that the free version only allows 2,000,000 characters to be translated per month per user.

[3] This corresponds to those used by the *baseline* system in WMT12-13. The list of features can be found on `http://www.quest.dcs.shef.ac.uk/quest_files/features_blackbox_baseline_17`.

| Method | # Training | # Test | MAE | RMSE |
|---|---|---|---|---|
| Mean | 1,881 | 9,000 | 0.151 | 0.201 |
| Prediction | 1,881 | 9,000 | 0.129 | 0.171 |

Table 1: Absolute score prediction: datasets and results for Fr-En

| Method | # Training | # Test | Kendall's $T$ |
|---|---|---|---|
| Random | 7,098 | 365 | 0.04 |
| Prediction | 7,098 | 365 | 0.16 |

Table 2: Ranking of alternative translations: datasets and results for De-En

with radial basis function (RBF) kernel was used as learning algorithm, since it has been shown to perform well in previous work (Callison-Burch et al., 2012; Bojar et al., 2013). The optimisation of parameters was done using grid search.

Both datasets are freely available. The French-English dataset is described in (Potet et al., 2012). It has 10,881 source sentences and their MT output and post-editions. We measure and estimate HTER scores between the MT and its post-edition. The first 1,881 sentences were used for training, and the rest for test. Performance was measured in terms of Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

The German-English dataset is provided by the WMT13 shared task on QE, with the official training/test splits used.[4] It has up to five alternative machine translations produced by different MT systems for each source sentence, which were ranked for quality by humans as part of the translation task in WMT08-WMT12. A baseline prediction model was trained using the rankings provided as absolute scores. This model was applied to each of the five alternative sentences and the predicted scores were used for ranking them. Performance was measured by comparing QuEst predictions to rankings performed by humans in terms of Kendall tau's correlation.

Table 3 gives a comparison between online QuEst and its previous, offline version in term of cumulative response time for all sentence pairs in each our two test sets along with the sizes of the language resources used to extract features for each of them. These figures include the time for loading models and sentences and input/output processing. Offline QuEst needs to do this for each sentence pair, while the new version loads all models only once, clearly showing better performance.

| Dataset | BING | FE | PR | FE + PR |
|---|---|---|---|---|
| Fr-En | 1.1 | 1.12 | 1.17 | 2.29 |
| De-En | 1.1 | 2.10 | 1.51 | 3.61 |

Table 4: Response time in seconds – per sentence – with an online interface of various models (FE = Feature Extractor, PR = Prediction)

We have also tested the response time of these pre-built models for each module in online QuEst, as shown in Table 4. These figures refer to running QuEst at a local host on a single core of machine Intel(R) Xeon(R) CPU E5-2620 0 @ 2.00GHz with 190GB of RAM. The response time for remote requests will depend upon the network speed. It is important to note the difference between response time for each of the dataset: The use of larger resources to extract features yields overall slower response time.

## 5. Remarks

QuEst can be downloaded from `http://www.quest.dcs.shef.ac.uk/`. The Web interface can be accessed at `http://www.quest.dcs.shef.ac.uk/QuEstClient_v1/test.php`

## 6. Acknowledgements

This work has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreements no. 296347 (QTLaunchPad) and 2011.4.2-287688 (Matecate).

## 7. References

E. Avramidis. 2013. Sentence-level ranking with quality estimation. *Machine Translation*, 28:1–20.

N. Bach, F. Huang, and Y. Al-Onaizan. 2011. Goodness: a method for measuring machine translation confidence. In *ACL11*, pages 211–219, Portland.

O. Bojar, C. Buck, C. Callison-Burch, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *WMT13*, pages 1–44, Sofia.

C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia. 2012. Findings of the 2012 Workshop on Statistical Machine Translation. In *WMT12*, pages 10–51, Montréal.

Y. He, Y. Ma, J. van Genabith, and A. Way. 2010. Bridging SMT and TM with Translation Recommendation. In *ACL10*, pages 622–630, Uppsala.

[4] `http://www.statmt.org/wmt13/quality-estimation-task.html`

| Dataset | SRC-LM | TGT-LM | POS-LM | GIZA | NGRAM | Offline QuEst | Online QuEst |
|---|---|---|---|---|---|---|---|
| Fr-En | 40M | 35M | 348K | 16M | 8.6M | 2,021 (0.224) | 343 (0.038) |
| De-En | 2G | 1.1G | 980K | 73M | 438M | 110 (0.300) | 22 (0.060) |

Table 3: Sizes of resources and cumulative response time in minutes for QuEst offline vs QuEst online for all sentence pairs (and per sentence pair) on each of the two test sets. The resources used to extract the features are source and target 3-gram language models (SRC-LM and TGT-LM), part-of-speech tag source language model (POS-LM), source-target Giza++ lexical table (GIZA), and raw counts of 1-3 grams in a corpus of the source language (NGRAM).

F. Josef Och and H. Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

M. Potet, E. Esperana-Rodier, L. Besacier, and H. Blanchon. 2012. Collection of a large database of french-english smt output corrections. In *LREC12*, Istanbul, Turkey.

K. Shah, E. Avramidis, E Biçici, and L Specia. 2013a. Quest - design, implementation and extensions of a framework for machine translation quality estimation. *Prague Bull. Math. Linguistics*, 100:19–30.

K. Shah, T. Cohn, and L. Specia. 2013b. An Investigation on the Effectiveness of Features for Translation Quality Estimation. In *Proceedings of MT Summit XIV*, pages 167–174.

R. Soricut and A. Echihabi. 2010. Trustrank: Inducing trust in automatic translations via ranking. In *ACL11*, pages 612–621, Uppsala.

L. Specia, M. Turchi, N. Cancedda, M. Dymetman, and N. Cristianini. 2009. Estimating the Sentence-Level Quality of Machine Translation Systems. In *EAMT09*, pages 28–37, Barcelona.

L. Specia, D. Raj, and M. Turchi. 2010. Machine translation evaluation versus quality estimation. *Machine Translation*, 24(1):39–50.

L. Specia, K. Shah, J.G.C. de Souza, and T. Cohn. 2013. Quest - a translation quality estimation framework. In *51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, ACL-2013, pages 79–84, Sofia, Bulgaria.

L. Specia. 2011. Exploiting objective annotations for measuring translation post-editing effort. In *EAMT11*, pages 73–80, Leuven.

A. Stolcke. 2002. SRILM - An Extensible Language Modeling Toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 901–904, Denver, CO.