Multi-level Translation Quality Prediction with QUEST++

Lucia Specia, Gustavo Henrique Paetzold and **Carolina Scarton**
Department of Computer Science
University of Sheffield, UK
{l.specia,ghpaetzold1,c.scarton}@sheffield.ac.uk

Abstract

This paper presents QUEST++ , an open source tool for quality estimation which can predict quality for texts at word, sentence and document level. It also provides pipelined processing, whereby predictions made at a lower level (e.g. for words) can be used as input to build models for predictions at a higher level (e.g. sentences). QUEST++ allows the extraction of a variety of features, and provides machine learning algorithms to build and test quality estimation models. Results on recent datasets show that QUEST++ achieves state-of-the-art performance.

1 Introduction

Quality Estimation (QE) of Machine Translation (MT) have become increasingly popular over the last decade. With the goal of providing a prediction on the quality of a machine translated text, QE systems have the potential to make MT more useful in a number of scenarios, for example, improving post-editing efficiency (Specia, 2011), selecting high quality segments (Soricut and Echihabi, 2010), selecting the best translation (Shah and Specia, 2014), and highlighting words or phrases that need revision (Bach et al., 2011).

Most recent work focuses on sentence-level QE. This variant is addressed as a supervised machine learning task using a variety of algorithms to induce models from examples of sentence translations annotated with quality labels (e.g. 1-5 *likert* scores). Sentence-level QE has been covered in shared tasks organised by the Workshop on Statistical Machine Translation (WMT) annually since 2012. While standard algorithms can be used to build prediction models, key to this task is work of feature engineering. Two open source feature extraction toolkits are available for that: ASIYA[1] and QUEST[2] (Specia et al., 2013). The latter has been used as the official baseline for the WMT shared tasks and extended by a number of participants, leading to improved results over the years (Callison-Burch et al., 2012; Bojar et al., 2013; Bojar et al., 2014).

QE at other textual levels have received much less attention. Word-level QE (Blatz et al., 2004; Luong et al., 2014) is seemingly a more challenging task where a quality label is to be produced for each target word. An additional challenge is the acquisition of sizable training sets. Although significant efforts have been made, there is considerable room for improvement. In fact, most WMT13-14 QE shared task submissions were unable to beat a trivial baseline.

Document-level QE consists in predicting a single label for entire documents, be it an absolute score (Scarton and Specia, 2014) or a relative ranking of translations by one or more MT systems (Soricut and Echihabi, 2010). While certain sentences are perfect in isolation, their combination in context may lead to an incoherent document. Conversely, while a sentence can be poor in isolation, when put in context, it may benefit from information in surrounding sentences, leading to a good quality document. Feature engineering is a challenge given the little availability of tools to extract discourse-wide information. In addition, no datasets with human-created labels are available and thus scores produced by automatic metrics have to be used as approximation (Scarton et al., 2015).

Some applications require fine-grained, word-level information on quality. For example, one may want to highlight words that need fixing. Document-level QE is needed particularly for gisting purposes where post-editing is not an option.

[1] http://nlp.lsi.upc.edu/asiya/
[2] http://www.quest.dcs.shef.ac.uk/

Proceedings of ACL-IJCNLP 2015 System Demonstrations, pages 115–120,
Beijing, China, July 26-31, 2015.

For example, for predictions on translations of product reviews in order to decide whether or not they are understandable by readers. We believe that the limited progress in word and document-level QE research is partially due to lack of a basic framework that one can be build upon and extend.

QUEST++ is a significantly refactored and expanded version of an existing open source sentence-level toolkit, QUEST. Feature extraction modules for both word and document-level QE were added and the three levels of prediction were unified into a single pipeline, allowing for interactions between word, sentence and document-level QE. For example, word-level predictions can be used as features for sentence-level QE. Finally, sequence-labelling learning algorithms for word-level QE were added. QUEST++ can be easily extended with new features at any textual level. The architecture of the system is described in Section 2. Its main component, the feature extractor, is presented in Section 3. Section 4 presents experiments using the framework with various datasets.

2 Architecture

QUEST++ has two main modules: a feature extraction module and a machine learning module. The first module is implemented in Java and provides a number of feature extractors, as well as abstract classes for features, resources and preprocessing steps so that extractors for new features can be easily added. The basic functioning of the feature extraction module requires raw text files with the source and translation texts, and a few resources (where available) such as the MT source training corpus and source and target language models (LMs). Configuration files are used to indicate paths for resources and the features that should be extracted. For its main resources (e.g. LMs), if a resource is missing, QUEST++ can generate it automatically.

Figure 1 depicts the architecture of QUEST++ . *Document* and *Paragraph* classes are used for document-level feature extraction. A *Document* is a group of *Paragraphs*, which in turn is a group of *Sentences*. *Sentence* is used for both word- and sentence-level feature extraction. A *Feature Processing Module* was created for each level. Each processing level is independent and can deal with the peculiarities of its type of feature.

Machine learning QUEST++ provides scripts to interface the Python toolkit `scikit-learn`[3] (Pedregosa et al.,). This module is independent from the feature extraction code and uses the extracted feature sets to build and test QE models. The module can be configured to run different regression and classification algorithms, feature selection methods and grid search for hyper-parameter optimisation. Algorithms from `scikit-learn` can be easily integrated by modifying existing scripts.

For word-level prediction, QUEST++ provides an interface for `CRFSuite` (Okazaki, 2007), a sequence labelling C++ library for Conditional Random Fields (CRF). One can configure CRFSuite training settings, produce models and test them.

3 Features

Features in QUEST++ can be extracted from either source or target (or both) sides of the corpus at a given textual level. In order describe the features supported, we denote:

- S and T the source and target *documents*,
- **s** and **t** for source and target *sentences*, and
- s and t for source and target *words*.

We concentrate on MT system-independent (*black-box*) features, which are extracted based on the output of the MT system rather than any of its internal representations. These allow for more flexible experiments and comparisons across MT systems. System-dependent features can be extracted as long they are represented using a predefined XML scheme. Most of the existing features are either language-independent or depend on linguistic resources such as POS taggers. The latter can be extracted for any language, as long as the resource is available. For a pipelined approach, predictions at a given level can become features for higher level model, e.g. features based on word-level predictions for sentence-level QE.

3.1 Word level

We explore a range of features from recent work (Bicici and Way, 2014; Camargo de Souza et al., 2014; Luong et al., 2014; Wisniewski et al., 2014), totalling 80 features of seven types:

Target context These are features that explore the context of the target word. Given a word t_i in position i of a target sentence, we extract: t_i,

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

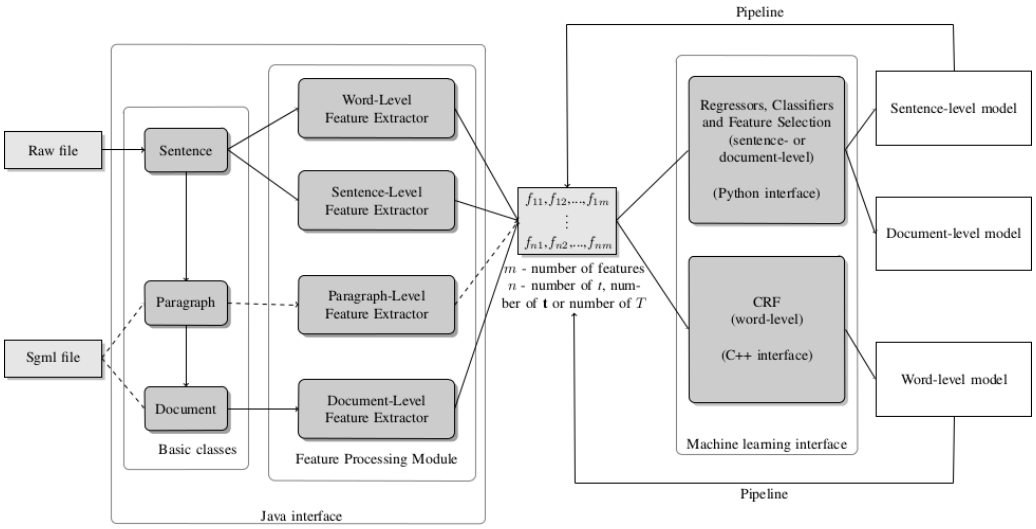

Figure 1: Architecture of QUEST++

i.e., the word itself, bigrams $t_{i-1}t_i$ and t_it_{i+1}, and trigrams $t_{i-2}t_{i-1}t_i$, $t_{i-1}t_it_{i+1}$ and $t_it_{i+1}t_{i+2}$.

Alignment context These features explore the word alignment between source and target sentences. They require the 1-to-N alignment between source and target sentences to be provided. Given a word t_i in position i of a target sentence and a word s_j aligned to it in position j of a source sentence, the features are: the aligned word s_j itself, target-source bigrams $s_{j-1}t_i$ and t_is_{j+1}, and source-target bigrams $t_{i-2}s_j$, $t_{i-1}s_j$, s_jt_{i+1} and s_jt_{i+2}.

Lexical These features explore POS information on the source and target words. Given the POS tag Pt_i of word t_i in position i of a target sentence and the POS tag Ps_j of word s_j aligned to it in position j of a source sentence, we extract: the POS tags Pt_i and Ps_j themselves, the bigrams $Pt_{i-1}Pt_i$ and Pt_iPt_{i+1} and trigrams $Pt_{i-2}Pt_{i-1}Pt_i$, $Pt_{i-1}Pt_iPt_{i+1}$ and $Pt_iPt_{i+1}Pt_{i+2}$. Four binary features are also extracted with value 1 if t_i is a stop word, punctuation symbol, proper noun or numeral.

LM These features are related to the n-gram frequencies of a word's context with respect to an LM (Raybaud et al., 2011). Six features are extracted: lexical and syntactic backoff behavior, as well as lexical and syntactic longest preceding n-gram for both a target word and an aligned source word. Given a word t_i in position i of a target sentence, the lexical backoff behavior is calculated as:

$$f(t_i) = \begin{cases} 7 & \text{if } t_{i-2}, t_{i-1}, t_i \text{ exists} \\ 6 & \text{if } t_{i-2}, t_{i-1} \text{ and } t_{i-1}, t_i \text{ exist} \\ 5 & \text{if only } t_{i-1}, t_i \text{ exists} \\ 4 & \text{if } t_{i-2}, t_{i-1} \text{ and } t_i \text{ exist} \\ 3 & \text{if } t_{i-1} \text{ and } t_i \text{ exist} \\ 2 & \text{if } t_i \text{ exists} \\ 1 & \text{if } t_i \text{ is out of the vocabulary} \end{cases}$$

The syntactic backoff behavior is calculated in an analogous fashion: it verifies for the existence of n-grams of POS tags in a POS-tagged LM. The POS tags of target sentence are produced by the Stanford Parser[4] (integrated in QUEST++).

Syntactic QUEST++ provides one syntactic feature that proved very promising in previous work: the Null Link (Xiong et al., 2010). It is a binary feature that receives value 1 if a given word t_i in a target sentence has at least one dependency link with another word t_j, and 0 otherwise. The Stanford Parser is used for dependency parsing.

Semantic These features explore the polysemy of target and source words, i.e. the number of senses existing as entries in a WordNet for a given target word t_i or a source word s_i. We employ the Universal WordNet,[5] which provides access to WordNets of various languages.

[4] http://nlp.stanford.edu/software/lex-parser.shtml

[5] http://www.lexvo.org/uwn/

Pseudo-reference This binary feature explores the similarity between the target sentence and a translation for the source sentence produced by another MT system. The feature is 1 if the given word t_i in position i of a target sentence S is also present in a pseudo-reference translation R. In our experiments, the pseudo-reference is produced by Moses systems trained over parallel corpora.

3.2 Sentence level

Sentence-level QE features have been extensively explored and described in previous work. The number of QUEST++ features varies from 80 to 123 depending on the language pair. The complete list is given as part of QUEST++ 's documentation. Some examples are:

- number of tokens in **s** & **t** and their ratio,
- LM probability of **s** & **t**,
- ratio of punctuation symbols in **s** & **t**,
- ratio of percentage of numbers, content-/non-content words, nouns/verbs/etc in **s** & **t**,
- proportion of dependency relations between (aligned) constituents in **s** & **t**,
- difference in depth of syntactic trees of **s** & **t**.

In our experiments, we use the set of 80 features, as these can be extracted for all language pairs of our datasets.

3.3 Document level

Our document-level features follow from those in the work of (Wong and Kit, 2012) on MT evaluation and (Scarton and Specia, 2014) for document-level QE. Nine features are extracted, in addition to aggregated values of sentence-level features for the entire document:

- content words/lemmas/nouns repetition in S/T,
- ratio of content words/lemmas/nouns in S/T,

4 Experiments

In what follows, we evaluate QUEST++'s performance for the three prediction levels and various datasets.

4.1 Word-level QE

Datasets We use five word-level QE datasets: the WMT14 English-Spanish, Spanish-English, English-German and German-English datasets, and the WMT15 English-Spanish dataset.

Metrics For the WMT14 data, we evaluate performance in the three official classification tasks:

- **Binary:** A Good/Bad label, where Bad indicates the need for editing the token.
- **Level 1:** A Good/Accuracy/Fluency label, specifying the coarser level categories of errors for each token, or Good for tokens with no error.
- **Multi-Class:** One of 16 labels specifying the error type for the token (mistranslation, missing word, etc.).

The evaluation metric is the average F-1 of all but the Good class. For the WMT15 dataset, we consider only the Binary classification task, since the dataset does not provide other annotations.

Settings For all datasets, the models were trained with the CRF module in QUEST++ . While for the WMT14 German-English dataset we use the Passive Aggressive learning algorithm, for the remaining datasets, we use the Adaptive Regularization of Weight Vector (AROW) learning. Through experimentation, we found that this setup to be the most effective. The hyper-parameters for each model were optimised through 10-fold cross validation. The baseline is the majority class in the training data, i.e. a system that always predicts "Unintelligible" for Multi-Class, "Fluency" for Level 1, and "Bad" for the Binary setup.

Results The F-1 scores for the WMT14 datasets are given in Tables 1–4, for QUEST++ and systems that oficially participated in the task. The results show that QUEST++ was able to outperform all participating systems in WMT14 except for the English-Spanish baseline in the Binary and Level 1 tasks. The results in Table 5 also highlight the importance of selecting an adequate learning algorithm in CRF models.

System	Binary	Level 1	Multiclass
QUEST++	0.502	0.392	**0.227**
Baseline	**0.525**	**0.404**	0.222
LIG/BL	0.441	0.317	0.204
LIG/FS	0.444	0.317	0.204
FBK-1	0.487	0.372	0.170
FBK-2	0.426	0.385	0.230
LIMSI	0.473	–	–
RTM-1	0.350	0.299	0.268
RTM-2	0.328	0.266	0.032

Table 1: F-1 for the WMT14 English-Spanish task

4.2 Pipeline for sentence-level QE

Here we evaluate the pipeline of using word-level predictions as features for sentence-level QE.

System	Binary	Level 1	Multiclass
QUEST++	**0.386**	**0.267**	**0.161**
Baseline	0.299	0.151	0.019
RTM-1	0.269	0.219	0.087
RTM-2	0.291	0.239	0.081

Table 2: F-1 for the WMT14 Spanish-English task

System	Binary	Level 1	Multiclass
QUEST++	**0.507**	**0.287**	**0.161**
Baseline	0.445	0.117	0.086
RTM-1	0.452	0.211	0.150
RTM-2	0.369	0.219	0.124

Table 3: F-1 for the WMT14 English-German task

Dataset We use the WMT15 dataset for word-level QE. The split between training and test sets was modified to allow for more sentences for training the sentence-level QE model. The 2000 last sentences of the original training set were used as test along with the original 1000 dev set sentences. Therefore, word predictions were generated for 3000 sentences, which were later split in 2000 sentences for training and 1000 sentences for testing the sentence-level model.

Features The 17 QUEST++ baseline features are used alone (Baseline) and in combination with four word-level prediction features:

- count & proportion of Good words,
- count & proportion of Bad words.

Oracle word level labels, as given in the original dataset, are also used in a separate experiment to study the potential of this pipelined approach.

Settings For learning sentence-level models, the SVR algorithm with RBF kernel and hyperparameters optimised via grid search in QUEST++ is used. Evaluation is done using **MAE** (Mean Absolute Error) as metric.

Results As shown in Table 6, the use of word-level predictions as features led to no improvement. However, the use of the oracle word-level labels as features substantially improved the results, lowering the baseline error by half. We note that the method used in this experiments is the same as that in Section 4.1, but with fewer instances for training the word-level models. Improving word-level prediction could thus lead to better results in the pipeline for sentence-level QE.

System	Binary	Level 1	Multiclass
QUEST++	**0.401**	**0.230**	**0.079**
Baseline	0.365	0.149	0.069
RTM-1	0.261	0.082	0.023
RTM-2	0.229	0.085	0.030

Table 4: F-1 for the WMT14 German-English task

Algorithm	Binary
AROW	**0.379**
PA	0.352
LBFGS	0.001
L2SGD	0.000
AP	0.000

Table 5: F-1 for the WMT15 English-Spanish task

	MAE
Baseline	0.159
Baseline+Predicted	0.158
Baseline+Oracle	**0.07**

Table 6: MAE values for sentence-level QE

4.3 Pipeline for document-level QE

Here we evaluate the pipeline of using sentence-level predictions as features for QE of documents.

Dataset For training the sentence-level model, we use the English-Spanish WMT13 training set for sentence-level QE. For the document-level model, we use English-Spanish WMT13 data from the translation shared task. We mixed the outputs of all MT systems, leading to 934 translated documents. 560 randomly selected documents were used for training and 374 for testing. As quality labels, for sentence-level training we consider both the HTER and the Likert labels available. For document-level prediction, BLEU, TER and METEOR are used as quality labels (not as features), given the lack of human-target quality labels for document-level prediction.

Features The 17 QUEST++ baseline features are aggregated to produce document-level features (Baseline). These are then combined with document-level features (Section 3.3) and finally with features from sentence-level predictions:

- maximum/minimum predicted HTER or Likert score,
- average predicted HTER or Likert score,
- Median, first quartile and third quartile predicted HTER or Likert score.

Oracle sentence labels are not possible as they do not exist for the test set documents.

Settings For training and evaluation, we use the same settings as for sentence-level.

Results Table 7 shows the results in terms of MAE. The best result was achieved with the

baseline plus HTER features, but no significant improvements over the baseline were observed. Document-level prediction is a very challenging task: automatic metric scores used as labels do not seem to reliably distinguish translations of different source documents, since they were primarily designed to compare alternative translations for the *same* source document.

	BLEU	TER	METEOR
Baseline	0.049	0.050	0.055
Baseline+Doc-level	0.053	0.057	0.055
Baseline+HTER	0.053	**0.048**	0.054
Baseline+Likert	0.054	0.056	0.054
Baseline+Doc-level+HTER	0.053	0.054	0.054
Baseline+Doc-level+Likert	0.053	0.056	0.054

Table 7: MAE values for document-level QE

5 Remarks

The source code for the framework, the datasets and extra resources can be downloaded from `https://github.com/ghpaetzold/questplusplus`.

The license for the Java code, Python and shell scripts is BSD, a permissive license with no restrictions on the use or extensions of the software for any purposes, including commercial. For pre-existing code and resources, e.g., `scikit-learn`, their licenses apply.

Acknowledgments

This work was supported by the European Association for Machine Translation, the QT21 project (H2020 No. 645452) and the EXPERT project (EU Marie Curie ITN No. 317471).

References

N. Bach, F. Huang, and Y. Al-Onaizan. 2011. Goodness: a method for measuring MT confidence. In *ACL11*.

E. Bicici and A. Way. 2014. Referential Translation Machines for Predicting Translation Quality. In *WMT14*.

J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing. 2004. Confidence Estimation for Machine Translation. In *COLING04*.

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 SMT. In *WMT13*.

O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, and A. Tamchyna. 2014. Findings of the 2014 Workshop on SMT. In *WMT14*.

C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia. 2012. Findings of the 2012 Workshop on SMT. In *WMT12*.

J. G. Camargo de Souza, J. González-Rubio, C. Buck, M. Turchi, and M. Negri. 2014. FBK-UPV-UEdin participation in the WMT14 Quality Estimation shared-task. In *WMT14*.

N. Q. Luong, L. Besacier, and B. Lecouteux. 2014. LIG System for Word Level QE task. In *WMT14*.

N. Okazaki. 2007. CRFsuite: a fast implementation of Conditional Random Fields. `http://www.chokkan.org/software/crfsuite/`.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12.

S. Raybaud, D. Langlois, and K. Smali. 2011. This sentence is wrong. Detecting errors in machine-translated sentences. *Machine Translation*, 25(1).

C. Scarton and L. Specia. 2014. Document-level translation quality estimation: exploring discourse and pseudo-references. In *EAMT14*.

C. Scarton, M. Zampieri, M. Vela, J. van Genabith, and L. Specia. 2015. Searching for Context: a Study on Document-Level Labels for Translation Quality Estimation. In *EAMT15*.

K. Shah and L. Specia. 2014. Quality estimation for translation selection. In *EAMT14*.

R. Soricut and A. Echihabi. 2010. Trustrank: Inducing trust in automatic translations via ranking. In *ACL10*.

L. Specia, K. Shah, J. G. C. de Souza, and T. Cohn. 2013. Quest - a translation quality estimation framework. In *ACL13*.

L. Specia. 2011. Exploiting objective annotations for measuring translation post-editing effort. In *EAMT11*.

G. Wisniewski, N. Pcheux, A. Allauzen, and F. Yvon. 2014. LIMSI Submission for WMT'14 QE Task. In *WMT14*.

B. T. M. Wong and C. Kit. 2012. Extending machine translation evaluation metrics with lexical cohesion to document level. In *EMNLP/CONLL*.

D. Xiong, M. Zhang, and H. Li. 2010. Error detection for SMT using linguistic features. In *ACL10*.