QE Viewer: an Open-Source Tool for Visualization of Machine Translation Quality Estimation Results

Felipe Soares
University of Sheffield
Broomhall
Sheffield S10 2TG
United Kingdom
fs@felipesoares.net

Anna Zaretskaya
TransPerfect
Passeig de Gràcia, 11
Esc B 5-2
08007 Barcelona, Spain
azaretskaya@translations.com

Diego Bartolomé
TransPerfect
Passeig de Gràcia, 11
Esc B 5-2
08007 Barcelona, Spain
dbartolome@translations.com

Abstract

QE Viewer is a web-based tool for visualizing results of a machine translation quality estimation (QE) system. It allows users to see information on the predicted post-editing distance (PED) for a given file or sentence, and highlighted words that were predicted to contain MT errors. The tool can be used in a variety of academic, educational and commercial scenarios.

1 Introduction

This paper presents a web-based tool for visualization of machine translation quality estimation (QE) (Specia et al, 2018) results. The tool allows users to submit one or several bilingual files with machine translation (MT) output and see information about its estimated quality, namely the predicted post-editing distance (PED), the distribution of segments with different PED scores in the file, and general information about the file in terms of number of segments and word count. Most importantly, the tools also allows to see all segments in the file and highlights words that potentially contain MT errors.

The source code is available at: `https://github.com/soares-f/qe-viewer`.

2 Functionality of the Tool

2.1 Features

QE Viewer is an easy-to-use web interface that allows users to upload a bilingual file in TMXL[1] format (or multiple TMXL files in a .zip archive) that contains source segments and MT segments. The user also has the option to introduce a unique identifier for that submission, allowing possible integration with other MT workflows.

Once the file is submitted it is processed with the QE system and the user sees the results page that contains a sensitivity bar allowing the user to adjust the system's prediction; file-level information on total number of segments, total number of words, predicted average PED; a histogram representing the distribution of segments in the file with different level of predicted PED (Figure 1); segment-level information showing the predicted PED for each segment in the file, while also highlighting the words that, according to the QE system's prediction, contain MT errors (Figure 2).

2.2 Integration with Different QE Frameworks

The tool is system-agnostic which means that it can be integrated with any QE framework, given that a pre-defined API request can be met. For this purpose, we implemented a wrapper for the OpenKiwi framework (Kepler et al., 2019) that handles the requests.

3 Technical Implementation

QE Viewer is implemented in Flask,[2] which is a Python micro-framework for web development that allows quick development and at the same time it can scale up to complex applications. In addition to the Flask framework, we used SQLAlchemy[3] as the object relational mapper, since it can abstract the database operations handling and allows easy database migration and scaling.

[1] TXML format is an XML-based format of bilingual files used at TransPerfect.

[2] `https://github.com/pallets/flask`

[3] `https://www.sqlalchemy.org/`

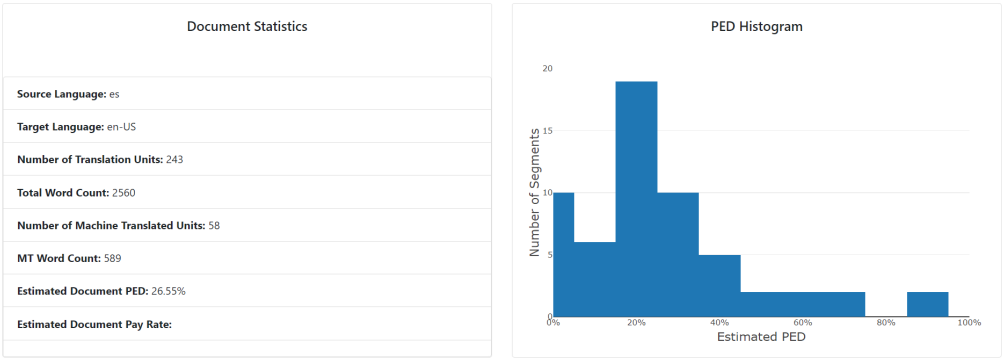

Figure 1: Histogram showing the distribution of segments with different PED.

ID	Source	Machine Translated	PE Dist.
1	ARE YOU COMPLYING	¿ESTÁ CUMPLIENDO	93.75%
2	WITH THE LIFE SAVING RULES ?	CON LAS REGLAS PARA SALVAR VIDAS ?	17.65%

Figure 2: Segment-level information.

When a file is submitted, the system checks if it is a ZIP file or a single TMXL file. Then the files are validated and a record is created both in the submission table and in the file information table. Once this step is done, the system will iterate over all available files for that submission and their segments, submitting calls to the QE system. The results are then stored in a table, such that the QE API will not need to be queried again when visualizing the same submission.

When visualizing, the system will look for the information stored in the corresponding tables and render the final page.

Regarding style and template, we employed Bootstrap[4] (v4.3) with the native Jinja2 templating system. Bootstrap is a responsive front-end component that can be easily modified to reflect the users' needs.

4 Conclusions

In this paper, we presented an open-source tool for visualization of QE results called QE Viewer. Its aim is to enable users to obtain and understand QE results in practical scenarios, such as translation industry or translation training. The tool is system-agnostic and can be integrated with any QE framework. QE Viewer has proven to be useful in a variety of commercial scenarios, including estimating the budget for post-editing, predicting MT errors and selecting the best MT system. In addition, it can be successfully used for educational purposes in translation and post-editing training.

[4] https://github.com/twbs/bootstrap

References

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

Kepler, Fabio, Trénous, Jonay, Treviso, Marcos, Vera, Miguel and Martins, André F. T.. 2019 OpenKiwi: An Open Source Framework for Quality Estimation. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Florence, Italy. 117–122.