

TRANSLATION QUALITY ASSURANCE WHAT IS MISSING? AND WHAT CAN BE DONE?

Dr. Hendrik J. KOCKAERT
Lessius/KULeuven
Department of Applied Language
Studies
Sint-Andriesstraat 2
B-2000 Antwerpen, BELGIUM

Prof. Dr. Frieda STEURS
Lessius/KULeuven
Department of Applied Language
Studies
Sint-Andriesstraat 2
B-2000 Antwerpen, BELGIUM

Julia MAKOUSHINA
Palex Languages & Software Ltd.,
ul. Uchebnaya 39/1
634034 Tomsk, RUSSIA

Abstract

This paper wants to capitalize on the recent attention on the growing need of impartial and quantifiable evaluation of translated documents. To achieve this, we need effective translation quality assurance (TQA) tools. A survey of the major currently available TQA tools has learnt us that TQA misses such an instrument. Many tools offer a quality control feature that automatically checks whether all the terms in the translation project match those in a pre-set terminology database. What most TQA tools have not offered yet is the functionality to run TQA processes beyond technical, consistency and metrics checking. A comparison of the characteristics and limitations of the major TQAs will show that terminology consistency and formal checking still remain the main features in present-day TQA tools. This paper's goal is to present a different TQA instrument, which is expected to tackle most of the limitations in computer assisted TQA, by blending the mere technical checks of every TQA with the translator's expertise and skilfulness in evaluating translations. This tool is expected to guarantee an overall consistency, transparency and impartiality in each revision, at all levels, ranging from punctuation to language register, or from collocational terminologies to style.

Key Words: translation quality assurance; automated and human quality management; correction memory

1. INTRODUCTION

Quality assurance (QA) is currently a vital step in the translation process in that it is necessary to ensure conformity to effective standards. What we normally understand under translation quality is that the translated text should be grammatically correct, have correct spelling and punctuation and sound as if it were originally written by a native speaker of the target language.

However, QA has yet another asset: assuring that the original text formatting has not been damaged, detecting unnecessary double spaces, or double full stops at the end of a sentence, verifying that the same sentences or translation segments have not been accidentally translated differently and that the project glossary has been followed, and numerous other tasks which do not require working knowledge of the target language.

Automation of translation processes is booming, and QA is not an exception. Usage of translation memories gives rise to the need of ensuring that the translator does not apply fuzzy matches as

perfect ones, makes the necessary corrections in the fuzzy matches, and warrants that only perfect matches supply the translation memory as such.

Apparently these tasks require a lot of attention, even thoroughness, and are therefore highly error-prone. It is obviously not easy to find a human translator who could produce fast and consistent results 8 hours a day, 5 days a week. Such tasks are relatively easy to formalize, hence to automate. As a result, all popular CAT tools introduce more automation features for QA/QC on every new release. Standalone QA/QC tools are also making fast progress.

In this paper, we wanted in the first place to gauge the stage at which current QA tools are able to assist (semi-)automated translation processes. We refer to Gerasimov (2007), Zetzsche (2007), and to the survey which was performed by ourselves (Makoushina, 2007) out of which we select the analysis results relevant for this paper. The QA tools we surveyed are listed in Table 1.

2. ANALYSIS OF CURRENT TQA TOOLS

Let us briefly consider what types of errors are normally covered by automated QA/QC tools [1].

Table 1: Comparative Analysis of Current TQAs

Check	Déjà Vu	SDLX QA Check	Star Transit	Trados QA Checker 2.0	Wordfast	ErrorSpy	QA Distiller	XBench
Segment-Level Checks								
Empty Translations	✓	✓		✓		✓	✓	✓
Forgotten Translations		✓		✓		✓	✓	
Skipped Translations				✓			✓	
Partial Translations		✓				✓	✓	
Incomplete Translations		✓		✓		✓	✓	
Corrupt Characters		✓		✓			✓	
Inconsistent Sentence Count		✓						
Inconsistency								
Source Inconsistency						✓	✓	✓
Target Inconsistency	✓	✓		✓		✓	✓	✓
Punctuation								
Punctuation at the End of Segments		✓		✓		✓	✓	
Spaces Before Punctuation		✓		✓		✓	✓	
Double Spacing		✓		✓	✓	✓	✓	
Double Dots		✓		✓		✓	✓	
Double Punctuation		✓				✓	✓	
Quotation Marks						✓	✓	
Brackets and Parentheses				✓		✓	✓	

Check	Déjà Vu	SDLX QA Check	Star Transit	Trados QA Checker 2.0	Wordfast	ErrorSpy	QA Distiller	XBench
Numbering								
Number Values	✓		✓	✓	✓	✓	✓	✓
Number Formatting			✓	✓		✓	✓	
Measurement Unit Conversion							✓	
Digit to Text Conversion						✓	✓	
Terminology								
Project Glossaries Adherence	✓	✓	✓		✓	✓	✓	✓
Identical Untranslatables						✓	✓	
Tags								
Identical Tags	✓		✓	✓		✓	✓	

3. RESULTS AND DISCUSSION

3.1 What is missing?

As we can observe in Table 1, Translation Quality Assurance (TQA) automation tools are currently capable to detect formatting or typesetting errors, and according to the benchmark tests, they are quite good at it and really help to save time and reduce translation costs while increasing the quality of translated documents. However, the tools normally do not support any type of linguistic checks. Advanced users are capable of formalizing some linguistic tasks by using regular expressions, but very often, staff responsible for translation quality assurance do not master sufficient technical knowledge for performing these tasks, and moreover, each error requires a separate preparation (i.e. separate regular expression), created manually. In this way, creating regular expressions mostly makes sense only for repetitive common errors, and, what is probably most important, is not “transparent” to the tools’ users. What is sure is that TQA tools do not replace human evaluators [2], but only help them.

When we further refer to the features offered by the TQA tools above, we can draw the following conclusions. Current TQAs expect the source documentation to be error-free, well authored, including terminological consistency, standardized spelling and subject matter experts' advice. TQAs automatically perform on the basis that all inconsistencies are equally incorrect, bypassing the fact that only standardized, client-based or in-house terminologies should be translated in a 100% exact match. More, general language segments often benefit from stylistic and linguistic variety to enhance readability and reader accessibility.

Subsequent to the above comparison's results, a needs analysis was performed in Lessius in order to determine which features we would really like to see added in an efficient and a more comprehensive TQA tool. The outcome was quite straightforward in that the main vacuum seemed to be the deficiency of a memory-based add-in device in an "evaluation memory" (EM) tool. Such a TQA would be able to house all the technical evaluation features, but also linguistic and stylistic corrections, comments, and even external links to bibliographical resources added by any evaluator. Next, it seemed to be advantageous to complement TQA tools by a statistics chart for the purpose of assisting the evaluator and/or the translator in the evaluation and translation processes respectively by showing error frequency and error types. Introducing statistics in a TQA tool will allow the end-user to benefit from a large scope translation flow including error data for

any subsequent translation or authoring task. Moreover, error statistics will prove useful for training reasons as well in a.o. tertiary institutes in translation studies, NATO and UN departments of translation, and any translation department which is concerned about improving QA.

3.2 What can be done?

Based on the results of the surveys and the needs analysis, we want to present in the next lines in which way a TQA tool can face the challenges that arise from the desiderata put forward by the surveyed translators and evaluators. What needs to be added to the current TQA features, seems to be a computer assisted process able to interact with, hence to partially supplement, intrinsic human activities of evaluation, not the least the chance of adding linguistic and stylistic comments in both source and target documentation, which are not automatically triggered by errors that are retrieved and handled in the current automation processes.

When comparing with the existing TQAs, the most stringent feature we need to add seems to be the opportunity of having a manual building up of corrections/comments by a human translator/evaluator. A device able to complement the above described automated evaluation features with manually added comments, which can be stocked in a database, is hoped to turn into a TQA tool we need. The Department of Applied Language Studies at Lessius and Palex Languages and Software have developed a design document of such an innovative device, and have baptised it as RevisionQ (RQ).

In the next lines, we portray a virtual TQA tool able to near this goal. The evaluator starts correcting one project, in which he builds up a revision/correction memory; for each next translation (within a similar domain and/or text type), the memory is built up, which is comparable to a Translation Memory (TM). The memory can be saved and re-used later by the same evaluator, an other evaluator, and even the same and/or another translator. Incorporating the main existing automated TQA features in RQ can be a plus, which will lead to a fusion of the above mentioned automated TQA features, and the manually built up database of earlier comments/suggestions. Based on the EM of an earlier translation, a window will pop up in the margin while moving the cursor over those segments that have already been evaluated earlier. The automated detection of the segments (items) where the evaluator has already made some comments is crucial: therefore, the EM is linked to the source text and the comments/suggestions already been entered. When RQ detects a segment for which a comment has been entered, it will display automatically the comments/suggestions the evaluator made during the evaluation of a previous translation of the text. In this way, RQ will guarantee an overall consistency in each revision, at all levels, ranging from punctuation to language register. It all depends on what the evaluator enters.

The evaluator will always be able to edit each comment/suggestion thanks to an Edit mode. What will be essentially useful in RQ, is the fact that each next revision of the same (similar) source text will generate exactly the same revision comments, hence a huge gain in evaluation consistency at all levels, technical, linguistic and stylistic. Thanks to the memory, each translator will receive an identical feedback, as broad as possible. Evaluating a number of similar texts will reduce enormously the very time-consuming task of evaluating translations, because the EM will always display each previously added comment, allowing to re-enter this comment at any time when needed. Regarding the evaluation itself, RQ will allow each translation project to be quoted in exactly the same way, which guarantees an objective and fully exchangeable evaluation scheme, and which harmonizes each evaluation which in turn leads to fairly quoted translations.

Technically, RQ will be able to count and display the errors automatically, and to display the marks in a separate window in the statistics chart area. Marking will be done based on an evaluation matrix set by each individual evaluator, or a default matrix. Each comment can be referenced exhaustively on the account that the evaluator needs to add all the necessary comments and references only once. Severity bench-marks based on international standards (EN, ISO, JA2450, ...) could also be applied to RQ. Thanks to the expertise developed by Palex, it will be possible to integrate RQ into their already advanced automated TQA. The EM will hopefully be linked to existing or new databases/TM's in order to ease further the workload. This will alleviate

at the same time the searching job for the exact terminology and phrases/context. In other words, external databases will help the evaluator and the translator choose the right terms/phrases before adding these to the RQ memory.

In addition to an evaluator's tool, the TQA tool is expected to use RQ as an assistant tool for translators while in the process of translating. Depending on default or customized access rights, RQ displays the previously checked comments that are valuable to the translators on the account of the evaluators' expertise, their previous search results, and advice from externally contacted domain experts.

4. SYSTEM OVERVIEW

Let us now descend to reality and present the implementable features at this stage of the development. The objective of this TQA project is to deliver a system that can be used to “capture” changes and corrections the evaluator makes in source and target documentation, to add comments for new users and to accumulate such information in a database for re-use. With each new segment in a text, the EM database is scanned for a similar record, and if one or more are detected, the information is retrieved and made available to each user involved in the translation process.

The following diagram shows how these components fit together:

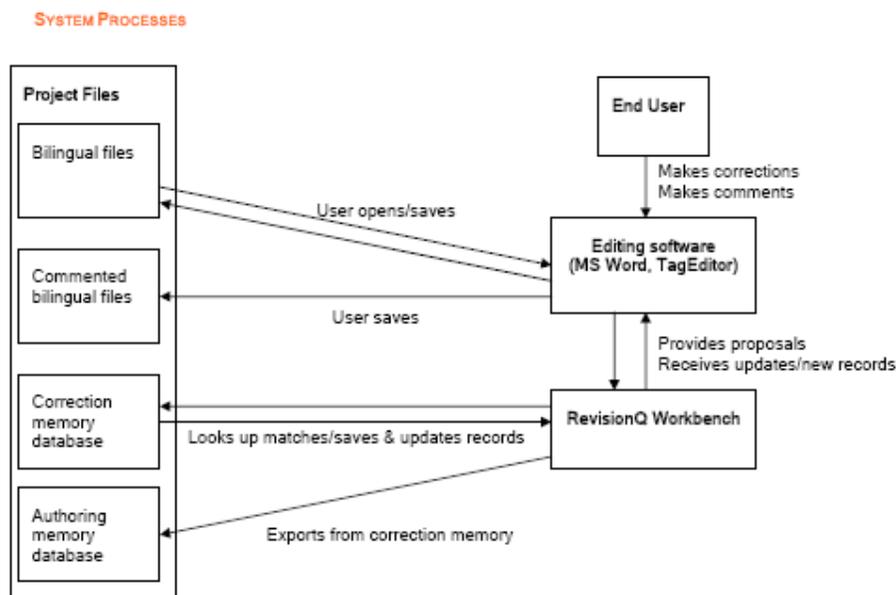


Figure 1: Palex QA and RevisionQ System process

4.1 RevisionQ Workbench

With the RevisionQ coupled to SDL Trados Workbench, the engine of the system is designed to manage with adequate success the EM database. The functions of the application are the following:

Open/close correction and authoring memories; Import/export correction and authoring memories; Populate correction memory by adding (or changing) records of source, target and corrected segments into the database (as well as any comments made by the evaluator) after the segment closure; Populate authoring memory by adding (or changing) records of source and corrected segments into the database after confirmation; Look up segments in the correction/authoring

memory based on user selectable criteria and visualize the options to the user. This means the user will see the previously entered corrections/comments on similar segments when translating a new (similar) text; Export partial data from correction memory to authoring memory; Search correction and authoring memories for user-specified text (concordance search).

Its main window might look as follows: the commented segments may be highlighted in *Source text*, *Original translation*, and in *Corrected translation* boxes. It is in this window that the "scanning or detecting" (by cursor gliding over segments) of translatable (or pre-translated by SDL Trados) segments in the *Source text*, while the translator is performing the translation of these segments in the *Original translation* window, will trigger the pop-up in the *Comments* box of comments/corrections that were already added by the evaluator in previous translations.

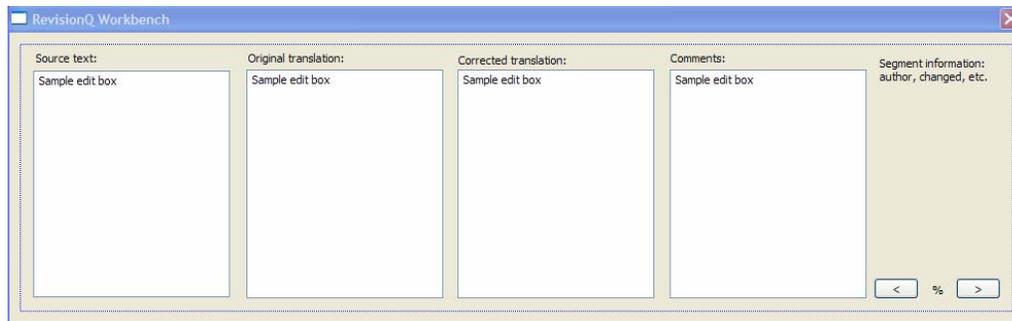


Figure 2: RevisionQ Workbench User Interface

The user interface will show four panes in a window. The buttons (< and >) on the right allow to switch between different matches with higher or lower percentage of similarity. The user will be able to switch different views in future versions (this is not the high priority feature). The information about the database record is shown above the buttons: author and time of segment translation, author and time of modifications, and file names.

This RevisionQ Workbench is one possible format example, and will be complemented by MS Word and TagEditor formats [3]. Let us take a glance at the the MS Word format.

4.2 RevisionQ MS Word

In MS Word, the evaluation macro set allows users to open source and target texts, to correct translation and to add comments to the file before finally sending the evaluation data to the database management program.

In MS Word, RQ works in a Track Changes mode where the evaluator inserts comments and corrections, and references to terminological databases.

The MS Word macro set will provide the user interface to the memory. It should work approximately as the SDL Trados Word macro set. The macro switches consequently segments of previously translated bilingual text and: Checks the open EM to find segments with the similarity percentage specified in the Options; Checks if any "terms" saved in the memory are found in the open segment; Provides both perfect/fuzzy matches and "term" matches to the user; Shows Track Changes the user adds in the translation. As soon as the user gets to the next segment, the macro set saves the changes and/or comments in the EM. If the same segment with other corrections already exists in the EM, it should behave according to the settings (overwrite after allowing sufficient access rights, keep the memory version, depending on the users' access rights).

5. CONCLUSION

Today, the above design document allows us to predict how the final product will eventually operate. Humanly entered add-ins are obviously not fully automated, but are a substantial part of the entire evaluation process, and are indeed valuable material that will undoubtedly benefit future translations. The fusion of human and computer-aided evaluation processes in a correction memory manager is expected to deliver a more comprehensive TQA tool for translators, evaluators and proofreaders than experienced at the present time.

6. NOTES

[1] This table refers only to the checks that may be enabled by selecting options in the tool interface. Most tools allow for expanding the number of supported checks via macros, SQL queries or regular expressions; however, those features are used rarely enough.

[2] In this paper, the evaluator's job refers, accordingly to EN 15038 (2006), to both reviews and revisions.

[3] The TagEditor plug-in allows users to open bilingual texts in .ttx format, to add changes and comments to segments and to send the evaluation data to the database management program. A Project file set contains all files associated with project: bilingual commented files (.ttx with comments for TagEditor 2007 and higher); bilingual "clean" files [comments deleted to deliver to the end client, .rtf for MS Words, .ttx for TagEditor]; correction memory file(s); optional authoring memory files.

7. REFERENCES

[1] EN 15038. (2006) *Translation services - Service requirements*. Technical Committee CEN/BT/TF 138, European Committee for Standardization.

[2] Gerasimov, A. (2007) "A comparison of translation QA products". *MultiLingual*, 85 (18): 22-25.

[3] Makoushina, J. (2007) "Translation Quality Assurance Tools: Current State and Future Approaches". *Translating and the Computer Conference*. Proceedings of the 29th International Conference on Translating and the Computer.

[4] Zetzsche, J. (2007) *The Translator's Tool Box. A Computer Primer for Translators* (5.5).