Raw Output Evaluator, a Freeware Tool for Manually Assessing Raw Outputs from Different Machine Translation Engines

Michael Farrell
IULM University
Milan, Italy
michael.farrell@iulm.it

Abstract

Raw Output Evaluator is a freeware tool, which runs under Microsoft Windows. It allows quality evaluators to compare and manually assess raw outputs from different machine translation engines. The outputs may be assessed in comparison to each other and to other translations of the same input source text, and in absolute terms using standard industry metrics or ones designed specifically by the evaluators themselves. The errors found may be highlighted using various colours. Thanks to a built-in stopwatch, the same program can also be used as a simple post-editing tool in order to compare the time required to post-edit MT output with how long it takes to produce an *unaided human translation* of the same input text. The MT outputs may be imported into the tool in a variety of formats, or pasted in from the PC Clipboard. The project files created by the tool may also be exported and re-imported in several file formats. Raw Output Evaluator was developed for use during a postgraduate course module on machine translation and post-editing.

1 Introduction

Raw Output Evaluator (ROE) is a tool designed to allow students to compare the raw outputs from different kinds of machine translation engine (rule-based, statistical, neural or any other kind), both to each other and to other translations of the same source text, and carry out comparative *human* quality assessment using standard industry metrics or ones designed specifically by the evaluators themselves. The same program can also be used as a post-editing tool and, thanks to a built-in stopwatch, to compare the time required to post-edit MT output with how long it takes to produce an *unaided human translation*.
It was developed for use during the postgraduate Machine Translation and Post-Editing Course Module of the Master's Degree in Specialist Translation and Conference Interpreting at the International University of Languages and Media (IULM), Milan, Italy[1].

In the first edition of the course module, the students initially tried using an existing tool called PET (Aziz et al. 2012) but, like translation environment tools in general, it only allows you to examine one source text and one output (target text) at a time, and was therefore not suitable for many of the course module exercises and experiments. Commercial quality evaluation tools, such as TAUS Quality Dashboard[2], were also not taken into consideration for similar reasons. Moreover some of the students did not find PET to be particularly user friendly, and there were some issues with characters with diacritics (ASCII code>127). All the students ended up resorting to Microsoft Word files and Microsoft Excel spreadsheets, but naturally found them rather clumsy for the purpose.

2 Methods

I decided to develop a specific software tool for the second edition of the course module to make *human* quality evaluation and comparing raw machine translation (MT) outputs easier. I

1 Machine Translation and Post-Editing, Course Module Syllabus, International University of Languages and Media (IULM), Milan, Italy: https://bit.ly/2NDrWY2
2 www.taus.net/quality-dashboard-lp

Proceedings of the 40th Conference Translating and the Computer, pages 38–49,
London, UK, November 15-16, 2018.

chose the macro scripting language AutoHotkey[3] not because it is the best for the kind of application I had in mind, but because I have developed other programs with it in the past and am therefore very familiar with it.

3 Results

The resulting freeware tool may be downloaded from the Internet[4]. The best way to explain what it can be used for is to describe the activities and experiments that were done with it during the course module it was designed for.

3.1 Comparison of Free Online MT Systems

The aim of this activity is to compare four free online MT systems: PROMPT Online-Translator[5] (a hybrid rule-based/statistical MT system), Yandex Translate[6] (a statistical MT system), Google Translate[7] (a neural MT system), and DeepL[8] (a neural MT system). The students are expected to find some similarities between the outputs from PROMPT and Yandex, and some between those from Google Translate and DeepL. They are also expected to find neural MT output to be better quality than the other kinds (Wu et al., 2016) and rank DeepL output as the best (Isabelle and Kuhn, 2018).

First of all, the students run ROE by clicking the roe deer icon on the Windows Desktop. They then choose *New* from the *File* menu.

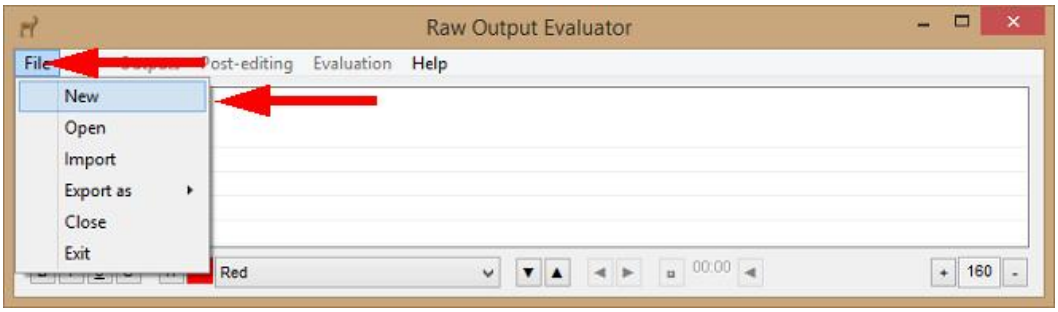

Figure 1: *File* menu

After that, they give a name to the ROE project file and click *Save*. The *Add Source Text* window then appears.

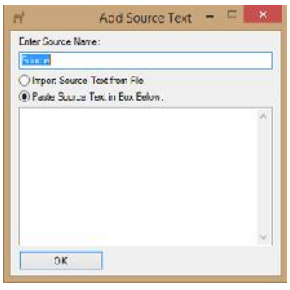

Figure 2: *Add Source Text* window

3 https://autohotkey.com
4 Raw Output Evaluator download: www.intelliwebsearch.com/raw-output-evaluator
5 www.online-translator.com
6 https://translate.yandex.com
7 https://translate.google.com
8 www.deepl.com/translator

The students leave the *Add Source Text* window open with the *Paste Source Text in Box Below* option selected, and choose a source text by opening the English language version of Wikipedia[9] in their browsers and picking an entry about a famous person. Each student should choose a different celebrity and select from 200 to 250 words, ideally from the biography section. For this reason it is best if they choose a dead person. ROE is not designed to be used with very long texts (max. 25 segments by default) and performs badly if loaded with excessive data; it is not intended for use by professional translators or post-editors, but as a teaching tool. For most classroom experiments and activities, short sample texts are in any case advisable.

The students copy their selected text to the Windows Clipboard (Ctrl+C), return to the *Add Source Text* window, paste in the text, and click *OK*. They then answer *Yes* to the question: *Ready to add an MT Output now?*

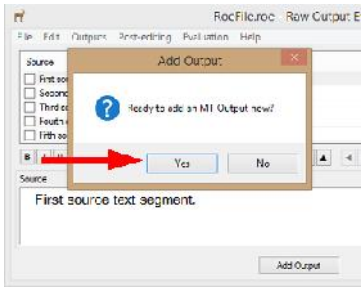

Figure 3: *Ready to add an MT Output now?*

At this point the *Add MT Output Text* window opens.

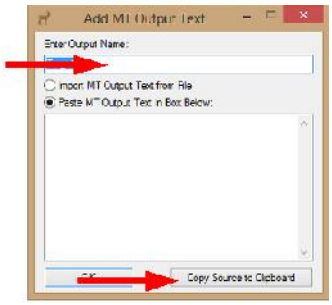

Figure 4: *Add MT Output Text* window

The students type the name of the first on-line MT engine they will use to translate the text (*PROMT*) into the *Enter Output Name* box, and click *Copy Source to Clipboard* to copy a *clean* version of the text to the Windows Clipboard. ROE automatically strips out various tags and extraneous characters to optimize the output of the MT engine. The students then leave the *Add MT Output Text* window open with the *Paste Source Text in Box Below* option selected, and open PROMPT Online-Translator in their browsers.

They paste the text from the Windows Clipboard into the left-hand box of PROMPT (Ctrl+V), set the source (English) and the target (Italian) languages, and click *TRANSLATE*. After that, they copy the whole Italian translation provided by PROMPT to the Windows Clipboard (Ctrl+C) and return to the *Add MT Output Text* window. At this point, they paste the text into the window (Ctrl+V) and click *OK*.

9 https://en.wikipedia.org

They then answer *Yes* to the question *Choose a QA Model?*

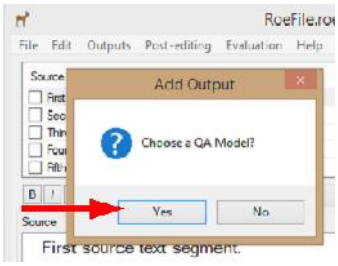

Figure 5: *Choose a QA Model?*

The students are advised to select *Best…Worst* from the *Non-Analytical Score* menu. Alternatively they may wish to give each segment a subjective score from 0 to 10 (*0…10*), decide that they pass or fail some subjective criteria (*Pass/Fail*) or simply decide how similar they are to one of the other outputs taken as a reference model (*Similarity*). If students wish to compare their results, they should all choose the same score system.

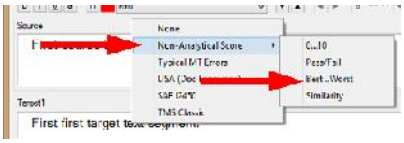

Figure 6: *Non-Analytical Score* menu

After that, they answer *Yes* to the question *Would you like to allow ties?* This allows them to give the same ranking to two different engines, i.e. joint best or joint worst.

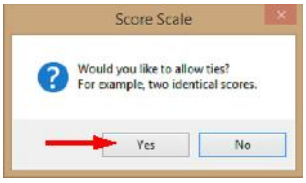

Figure 7: *Would you like to allow ties?*

They then click the *Add Output* button at the bottom of the user interface.

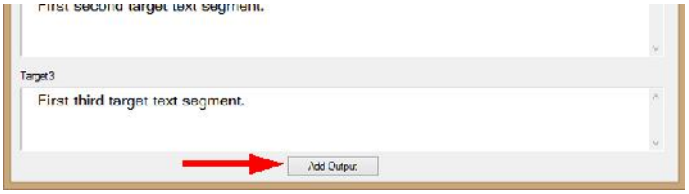

Figure 8: *Add Output* button

From this point onwards, they repeat the steps shown above for three more on-line MT engines:

- Yandex Translate

- Google Translate
- DeepL

In the end, they have:

- A segmented original English text.
- Four Italian translations of the same text from four different free on-line MT engines.

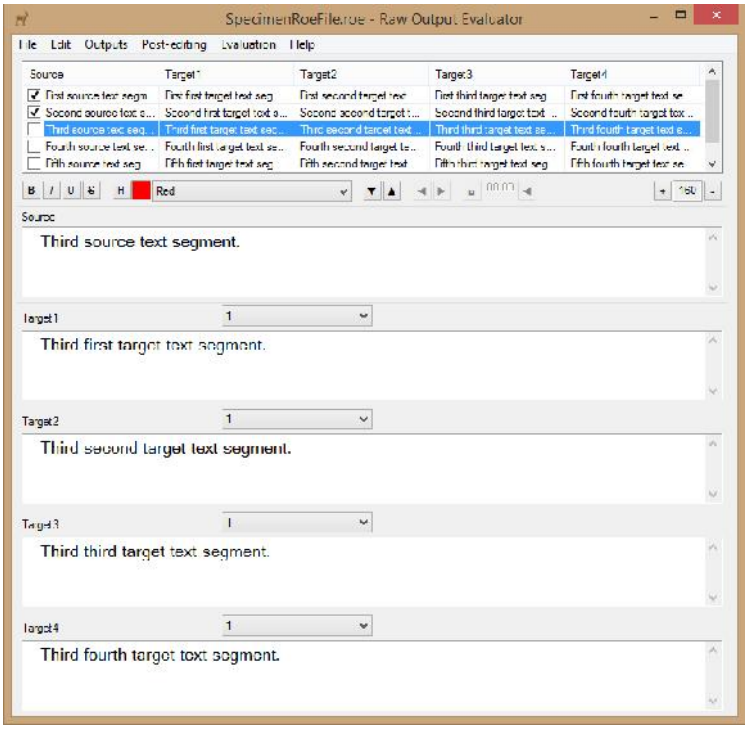

Figure 9: Interface with source text plus four target texts

If any segments are split incorrectly, the students can put them right by putting the cursor in the segment that needs fixing (in the case of *Split*, the cursor must be put precisely where the segment needs splitting) and choosing *Join* or *Split* from the *Edit* menu (Ctrl+J or Ctrl+S).

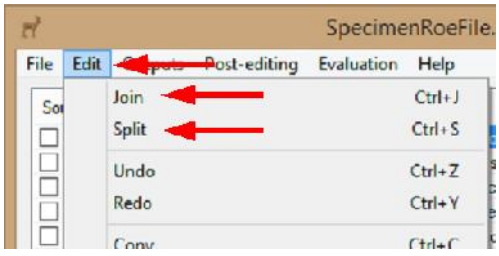

Figure 10: *Join* and *Split*

The tool then tells the user what the effects of the *Join* or *Split* will be and asks for confirmation.

The students then compare the four Italian MT outputs with the original English text, and with each other. They can highlight some of the most glaring errors with various colours to help with the evaluation.

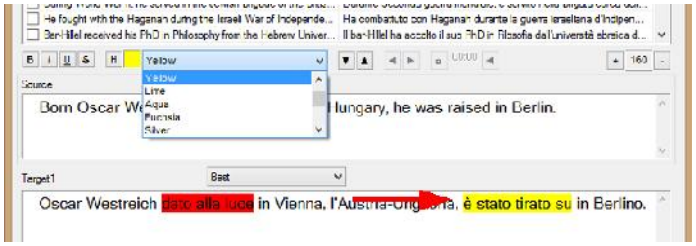

Figure 11: Error highlighting

They should then rate the various outputs (Best, Second Best, Third Best, etc.)

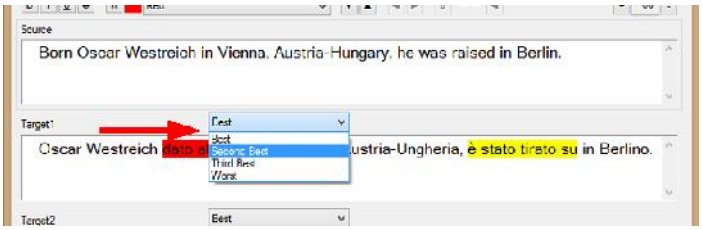

Figure 12: Segment rating

To move from one segment to the next, confirm the rating assigned, and save the highlighting colours, the students use Alt+Up / Alt+Down or the ▲▼ buttons. They may also confirm a rating and save the colours by clicking the check box on the left of the segment. Once all the segment ratings have been confirmed, they can calculate the total rating by choosing *Calculate Total Score* from the *Evaluation* menu.

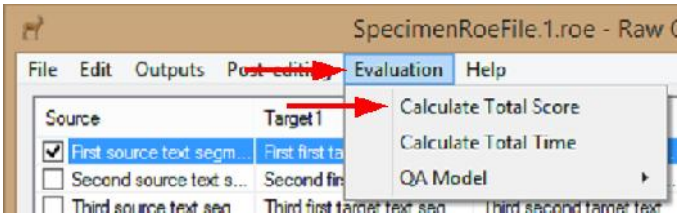

Figure 13: *Calculate Total Score*

The *Total Score* window opens.

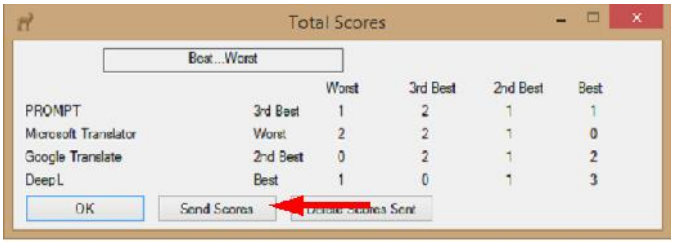

Figure 14: *Total Score* window

The classroom activity can stop here, or it can continue by calculating the overall rating for the whole class if all students have chosen the same rating system. However, to do this, the lecturer has to create a web app on the server to manage the data. An example app written in Classic ASP can be downloaded from the ROE help webpage[10]. If you create the app and set the web app URL in the ROE settings (Edit>Options), the students can then click *Send Scores* to send their ratings to the server. A window appears where they have to define the order of the MT engines so that the server adds the right scores together.

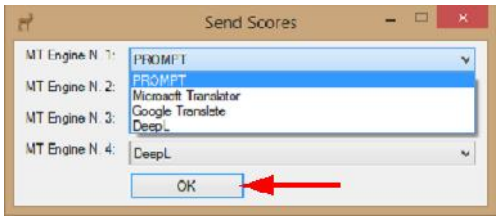

Figure 15: *Send Scores* window

If everything works, they see a *Score successfully processed* message and the lecturer's special web page displays the overall class rating (the page will need refreshing after the last student sends their ratings).

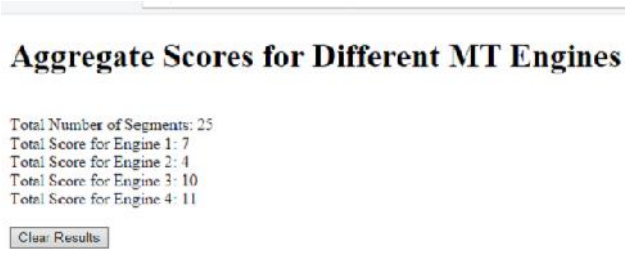

Figure 16: Overall scores webpage

A Microsoft PowerPoint presentation with instructions for this activity may be downloaded from the ROE help webpage[11] for use in the classroom.

3.2 Comparison of Translation and Post-Editing Times

In this activity, the students are divided into two groups. One group does *unaided human translations*, and the other post-edits MT output obtained from the same text. *Unaided* here means without a translation memory. However the students are allowed to use any dictionaries and web resources they wish, except MT. They compare the time taken to complete the two tasks. The built-in stopwatch feature makes this activity particularly simple.

After creating the ROE file, the students who do the *human* translation import the source text. This can be done by pasting the text in as described in the first activity above, or by importing a file (select the *Import Source Text from File* option on the *Add Source Text* window). The text may be imported from several kinds of file:

- Tabular files with any number of columns, such as:

10 www.intelliwebsearch.com/raw-output-evaluator-help/#faq-Calculatetotalaggregatescores
11 www.intelliwebsearch.com/raw-output-evaluator-help/#faq-ComparisonofFreeOnlineMTSystems

- Another Raw Output Evaluator project file (.roe)
- A standard comma separated file (.csv)
  - The field separator must be a comma, and not a semicolon or other character.
- A Microsoft Excel file (.xlsx and .xls)
  - Microsoft Excel must be installed on the PC.
  - The worksheet with the data to be imported must be the active one.

- A plain text file (.txt).
- Files which may be opened with Microsoft Word.
  - Microsoft Word must be installed on the PC.

ROE has been tested with Microsoft Word document files (.doc and .docx), Rich Text Format files (.rtf) and Hypertext Markup Language files (.htm and .html). In theory it should work with all file types Microsoft Word is able to read. In the case of a tabular file, the user has to choose the text column to import and indicate if the first row contains field names.

After the source text has been added, the students choose *Source* under the *Translate* item on the *Post-editing* menu.

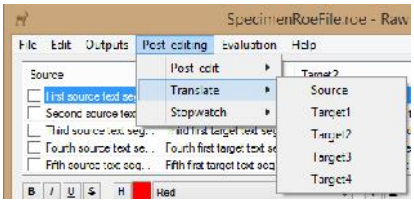

Figure 17: *Translate* menu

This creates a new empty column in ROE where the students should type their translations. When the new column is created, the user is asked if they would like to enable the stopwatch control buttons.

Figure 18: Stopwatch control buttons

The time is measured for each segment individually. The total time can be calculated by choosing *Calculate Total Time* from the *Evaluation* menu. The user may start, stop and reset the stopwatch for the segment displayed. If the stopwatch is not stopped before moving onto a new segment, it automatically stops in the old one and starts immediately in the new one. If the stopwatch is not stopped before resetting, it automatically starts again from zero immediately after reset.

After importing the source text, the post-editors import the raw MT output (called *Target1* by default). They then choose the *Target1* item under *Post-edit* on the *Post-editing* menu. This creates a duplicate text (called *Target2* by default) containing the same raw MT output, which the post-editors should then edit. The post-editors are also asked if they wish to enable the stopwatch controls.

3.3 Identifying MT Markers in Post-Edited MT Output

The students take the translations and post-edited texts created in the activity described in point 3.2 above and examine them to see if there are any MT markers (n-grams) which might be used to tell translation and post-edited MT output apart. ROE's text marking features make this activity particularly easy. The results and details of this experiment can be found in a separate paper (Farrell, 2018). The lecturer creates two new ROE project files for the students, one containing the source text, the raw MT output, and all the post-edited versions, and the other containing the source text and all the translations. There is no set limit to the number of texts that can be imported and compared in ROE, provided they are not too long, and it has been successfully used to examine 26 translations plus one source text all in the same file.

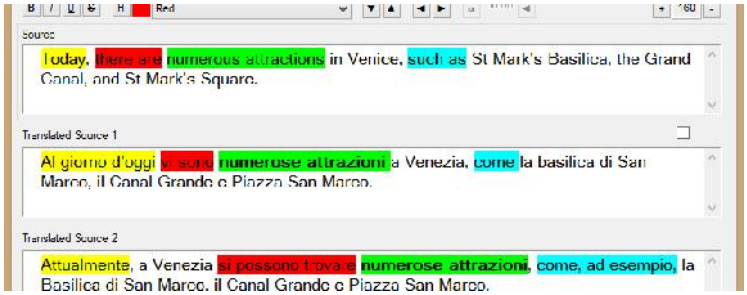

Figure 19: Comparing the translation of different n-grams

ROE can only display up to 5 texts at a time, usually one source text and four target texts. To display the other texts the students use the next/previous output display block buttons ◄►.

Figure 20: Next/previous output display block buttons

By default only the source text remains fixed (permanently displayed), and the others are replaced with the next or previous four target texts. However the lock check boxes can be used to prevent other texts from being replaced. This is useful in this particular activity to make sure that both the source text and the raw MT output are constantly displayed while the students compare the various post-edited versions of the same raw output.

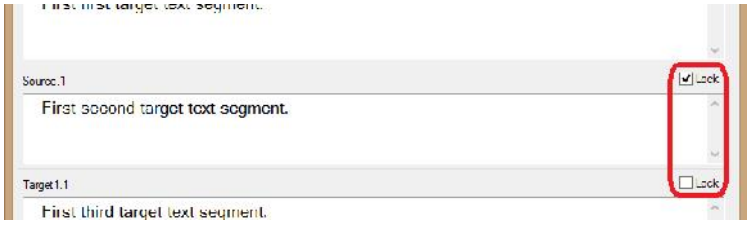

Figure 21: Segment lock check boxes

3.4 Quality Evaluation Metrics and Typical MT Errors

This activity is designed to teach the students to evaluate raw MT output manually using standard industry metrics, and to learn to identify the specific kinds of error typically found in raw MT output. The tool comes preset with *LISA (Doc Language)*, *SAE J2450* and *TMS*

Classic QA models. It is also possible to add a new QA model, or edit or delete an existing one. The *Typical MT Errors* QA model provided is based on the error types defined by Federico Gaspari in Gaspari et al. (2011) and completed with three types based on the observations of Esperança-Rodier et. al (2017) regarding unknown word errors. When one of the QA models is chosen, the highlighting colours correspond to one of the error categories defined in that model.

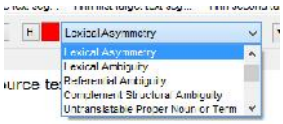

Figure 22: Error category highlighting colours

The user is also asked to set a pass/fail threshold when appropriate in terms of maximum error score permitted per n words, characters or segments.

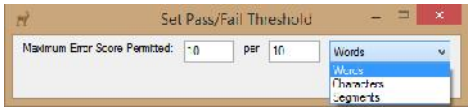

Figure 21: Pass/fail threshold setting window

When a segment is confirmed, some of the QA models require the students to complete an error questionnaire for each target text to summarize the errors highlighted and define their severity.

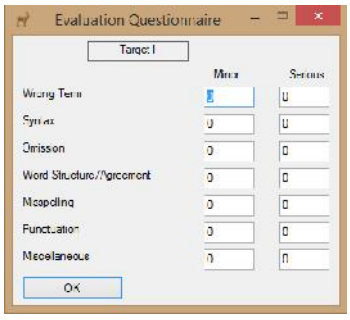

Figure 22: Error questionnaire

3.5 Evaluation of a Custom MT Engine

In the second semester of the course module, the students built a custom machine translation (CMT) engine with KantanMT. The experience is described in a separate paper (Farrell, 2017). In order to evaluate the output produced by their CMT engines, the students carried out several experiments, many of which were made simpler by using ROE.

To produce the material for their experiments, they took a text, for which there was an existing translation which had not been used as training data for the CMT engine (*official* text), and used it as input in three different tools:

- Their KantanMT CMT engine.
- Google Translate.

- A classic translation environment tool set up using the CMT engine training data corpus as a translation memory and only using the translation memory system features of the tool.

The raw output from each was then compared with the *official* text using ROE.

3.6 Other Tool Features

The user may create a new ROE project file (.roe) and populate it with data imported from various common file types used by CAT tools:

- XML Localisation Interchange File Format (XLIFF)
    - Source text plus one target text.
- Translation Memory eXchange (TMX)
    - Source text plus one target text.
- Standard comma separated file (.csv)
    - The field separator must be a comma, and not a semicolon or other character.
    - Source text plus up to four target texts.
- Microsoft Excel (.xlsx and .xls)
    - Source text plus up to four target texts.
    - Microsoft Excel must be installed on the PC.
    - Only the active worksheet is imported.

In the case of comma separated and Microsoft Excel files, the user is asked if the first row contains field names.

The user may also export data from the currently open ROE project file to various common file types used by CAT tools:

- XLIFF (XML Localisation Interchange File Format)
    - Source text plus up to four target texts.
- TMX (Translation Memory eXchange)
    - Source text plus up to four target texts.
- CSV (Comma separated file)
    - The user is asked to specify the field separator (comma, semicolon or tab).
    - Source text plus up to four target texts.

In the case of XML Localisation Interchange File Format and Translation Memory eXchange files, the user is asked to specify the languages of each output. In the case of comma separated files, the user is asked to specify the character used to separate the fields.

ROE is also able to mark and unmark segments which are identical to other parallel segments in a different text to see if two MT engines, two translators or two post-editors have come up with exactly the same translation.

Besides highlighting text with various colours, the user may also format it in bold, italics, strikethrough and underlining. Moreover it is possible to increase the display font size (zoom) for users with eyesight problems.

ROE is available in a Windows installation package, which allows the tool to be installed and uninstalled just like any other Windows program.

The user may also access on-line help (F1)[12] and check for program updates.

4 Discussion

ROE is not able to calculate the automatic metrics used to evaluate MT engine performance (BLEU, F-Measure, TER, etc.). This is not an issue for the course module it is designed for, since those metrics are automatically calculated by the CMT platform used in the second semester. However I am considering adding automatic quality evaluation capabilities. Rather than reinventing the wheel, I have studied the feasibility of integrating the Natural Language Toolkit, which runs under Python (Bird et al., 2009). It would seem to be implementable, but would make package installation rather more complicated. I have therefore put the project on hold while awaiting feedback from the academic community after the official launch of ROE through the publication of this paper.

5 Conclusions

The tool has greatly eased the difficulty of carrying out various activities and experiments during the course module it was designed for, thus allowing students to concentrate on acquiring knowledge about MT and post-editing. It can therefore be considered a success.

Acknowledgements

All trademarks and trade names are the property of their respective owners.

References

Aziz, Wilker, Sheila Castillo Maria de Sousa, and Lucia Specia (2012). PET: a Tool for Post-editing and Assessing Machine Translation. Proceedings of the 16th Annual Conference of the European Association for Machine Translation, pages 3982-3987.

Bird, Steven, Edward Loper and Ewan Klein (2009): Natural Language Processing with Python. O'Reilly Media Inc.

Esperança-Rodier, Emmanuelle, Caroline Rossi, Alexandre Bérard, Laurent Besacier (2017): Evaluation of NMT and SMT Systems: A Study on Uses and Perceptions. Proceedings of the 39th Conference Translating and the Computer, pages 11–24, London, UK, November 16-17, 2017.

Farrell, Michael (2017): Building a Custom Machine Translation Engine as part of a Postgraduate University Course: a Case Study. Proceedings of the 39th Conference Translating and the Computer, pages 35–39,
London, UK, November 16-17, 2017.

Farrell, Michael (2018): Machine Translation Markers in Post-Edited Machine Translation Output. Paper to be presented at the 40th Conference Translating and the Computer, London, United Kingdom, 15-16 November, 2018.

Gaspari, Federico, Guy Aston, Elena Di Bello, Claudia Lecci, Eros Zanchetta. Edited by Gabriele Bersani Berselli (2011), Usare la traduzione automatica, CLUEB Editrice, Bologna, Italy

Isabelle, Pierre and Roland Kuhn (2018): A Challenge Set for French --> English Machine Translation. ArXiv e-prints.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi (2016): Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. ArXiv e-prints.

12 www.intelliwebsearch.com/raw-output-evaluator-help