How do LSPs compute MT discounts? Presenting a company's pipeline and its use

Randy Scansani
Acolad
Rimini, Italy
rscansani@acolad.com

Lamis Mhedhbi
Acolad
Paris, France
lmhedhbi@acolad.com

Abstract

In this paper we present a pipeline developed at Acolad to test a Machine Translation (MT) engine and compute the discount to be applied when its output is used in production. Our pipeline includes three main steps where quality and productivity are measured through automatic metrics, manual evaluation, and by keeping track of editing and temporal effort during a post-editing task. Thanks to this approach, it is possible to evaluate the output quality and compute an engine-specific discount. Our test pipeline tackles the complexity of transforming productivity measurements into discounts by comparing the outcome of each of the above-mentioned steps to an estimate of the average productivity of translation from scratch. The discount is obtained by subtracting the resulting coefficient from the per-word rate. After a description of the pipeline, the paper presents its application on four engines, discussing its results and showing that our method to estimate post-editing effort through manual evaluation seems to capture the actual productivity. The pipeline relies heavily on the work of professional post-editors, with the aim of creating a mutually beneficial cooperation between users and developers.

1 Introduction

Over the last few years, the number of companies starting to integrate machine translation (MT) in their workflow has increased dramatically. In 2018, for the first time, more than half of the companies and individual professionals taking part in the Language Industry Survey stated that they used MT (Elia et al., 2018). The same survey repeated in 2019 showed that MT was one of the highest priorities for companies, with 51% of them willing to increase its use and 62% stating that they were planning investments on MT (Elia et al., 2019).

At the same time, all categories involved in the 2019 Language Industry Survey mentioned price pressure as the main negative trend, with MT and post-editing identified as one of the major causes. In the previous year, one of the main concerns of the language industry components were the "technological advances that are not initiated or controlled by the respondents" (Elia et al., 2018, p. 31).

These surveys point out that individuals and companies are embracing a technology that they themselves see as a threat to the sustainability of the translation industry. This seems to suggest that there might be a strong disagreement regarding the price models to be adopted when MT is included in the workflow. One of the priorities of MT users and developers should therefore be the creation of shared models and/or methods to measure quality and productivity with a view to computing discounts to be applied when MT is used. However, to the best of our knowledge this topic has been underinvestigated so far.

The "Pricing Machine Translation Post-editing Guidelines" published by TAUS have tried to fill the gap in this field.[1] They claim that a model to price post-editing should be predictive (able to es-

[1] TAUS is an industry organization in the field of translation and languages. The guidelines can be found here: https://bit.ly/2PlTCUd.

timate prices in advance), fair (involving all parties and providing them with a reliable estimate), and appropriate for both the language pair and the content characteristics. To achieve this aim, three main indicators should be combined: human evaluation, automatic metrics, and productivity assessments. Despite the usefulness of these guidelines, many questions on how to convert quality scores and/or judgements into rates – especially word-based ones – or discounts remain unanswered. For example, if automatic metrics are used, what would a TER of 0.4 imply in terms of productivity? If a sentence is evaluated as very fluent by a human translator, how could this judgement be converted into a discount? At the same time, if productivity assessments on a machine translated sample reveals that, e.g., 200 words per hour were post-edited, what would be the most reliable and consistent way to turn this productivity rate into a fair per-word rate for post-editing? Indeed, measuring post-editing productivity might be a difficult task *per se*, since post-editing effort is multi-dimensional and composed of time needed to edit a text (temporal effort), number of edits performed (editing effort) and identification and correction of issues (cognitive effort) (Krings, 2001). Computing discounts adds an additional challenge, i.e. converting productivity measures into post-editing rates.

Building upon TAUS guidelines, in the present work we introduce a pipeline to estimate an engine-specific discount to be applied to per-word rates for post-editing. The pipeline is composed of different steps that involve the use of automatic metrics, post-editing effort measurements and human evaluation processes, carried out as follows:

1. Automatic scores. TER (Snover et al., 2006) is used to evaluate the engine and gain a first insight into its quality (see Section 3.1).

2. Manual evaluation. One linguist is asked to manually evaluate and score, based on the amount of editing required, each sentence of a sample of 3,000 words (see Section 3.2).

3. Real Condition test (RCT). One post-editor is asked to perform full post-editing on a sample of 5,000 words. Both editing and temporal effort are measured (see Section 3.3).

To tackle the aforementioned challenge of converting quality scores and productivity rates into discounts for per-word rates, the outcomes of the second and third steps are contrasted against what would be the average productivity of a linguist translating from scratch. The resulting coefficient is then subtracted from the per-word rate applied to no-match sentences (see Section 3.4). A high coefficient thus results in a low discount and *vice versa*.

Our model complies with the three requirements listed by TAUS. It computes discounts in advance, and being engine-specific it provides a discount that is appropriate for a content type and a language pair. Also, the Acolad pipeline involves all parties, since it strongly relies on the work of post-editors who perform the tasks listed above (and described in Section 3.2 and 3.3) and fill in a feedback module on the output quality at the end of each post-editing task. Besides being desirable according to TAUS guidelines and useful to better understand an engine quality, involving post-editors allows us to collect data coming from the final users of our custom engines and to raise awareness of our MT-related processes, especially those focused on MT quality, which might help translators to feel more comfortable when performing post-editing tasks.

The present work is structured as follows. Section 2 provides a brief overview on articles and lines of research that are relevant for the topic handled here. Each stage of our pipeline is detailed in Section 3 and its subsections, while the pipeline application is presented in Section 4. To conclude, results of the application are discussed in Section 5, and limitations and future work are outlined in Section 6.

2 Related work

Quality estimation (QE) models – i.e. models trained to output a quality indicator of a translation without the need of comparing it to a reference – have been examined by a large amount of papers in the last decade and would serve the need of estimating the output quality, the post-editing time or the amount of editing needed on a whole text or on each of its sentences (Specia et al., 2009; Escartín et al., 2017; Scarton et al., 2019). Despite the relevance of this line of research for the present paper, we are not aware of QE models aimed at predicting discounts or post-editing effort that are ready to be integrated in CAT tools, which would be key for translation companies. One exception is

Memsource MTQE functionality, which computes a match rate for machine translated sentences – similarly to what CAT tools usually do for translation memory (TM) matches. However Memsource MTQE is still in a Beta version, and as of today it is not able to estimate quality for all language pairs and sentences.[2]

Companies operating in the translation field have published articles on their experience in computing discounts for post-editing. Cattelan (2013) describes an experiment carried out in the framework of the MateCat project aimed to understand the maximum discount translators are willing to accept for post-editing tasks.[3] Translators working on different language combinations were asked to choose between translating 1,000 words from scratch or post-editing a lower number of words at the same per-word rate. The number of words paid for post-editing was gradually increased until at least 75% of the translators chose post-editing over translation from scratch. However, the author claimed that this method was not suitable for establishing discount rates, since for some language combinations translators chose post-editing over translation only when the former was paid more than the latter. The article then suggests a combination of editing effort, temporal effort and output quality to understand the usefulness of MT for post-editors.

A rather straightforward approach at computing post-editing discounts is the one presented by Lizuka (2018) in an SDL blog post.[4] The author suggests that a post-editing test is set up where a linguist is asked to post-edit a text sample for one hour. After that, the post-editor hourly rate is divided by the number of translated words to obtain the discount. However, since post-editing rates should be computed based on the content and language pair (see Section 1), this test should be carried out every time a post-editor is working on a new content, or on a text belonging to a different domain or on a target language they have never post-edited. This would not be the most efficient solution.

[2]A description of the Memsource MTQE feature can be found at this link: `https://bit.ly/2u1exEO`, while the supported language pairs are listed here `https://bit.ly/2wCHDuR`.

[3]MateCat is a free online CAT tool developed in the framework of a three-year project funded by the European Union in 2011.

[4]SDL is a company offering translations services and software.

The different approaches presented here suggest that developing a reliable method to compute post-editing discounts is not an easy task. As a matter of fact, to the best of our knowledge, as of today the translation world lacks a shared model to establish MT discounts upfront. In the next sections we will present the pipeline developed and currently in use at Acolad to compute an engine-specific discount (see Section 3). To gain a better insight into its functioning, four use cases are also presented (see Section 4).

3 Pipeline description

Our pipeline is mainly composed of the three steps that were introduced in Section 1. These will be detailed in the following sections, together with a final step in which the final coefficient is computed.

3.1 Automatic scores

After each (re-)training, our MT engine quality is measured based on TER on a held-out set of 2,500 sentence pairs. This step is carried out for two main reasons. First, it produces a quality indicator whose results are comparable across different versions of the engine. Moreover it provides a first insight into the output quality. If TER results are not satisfactory, tests described in Section 3.2 and 3.3 are not launched. Instead, analyses are carried out on the hypothesis to understand the reason for such poor quality.

3.2 Manual evaluation

As mentioned in Section 1, including a manual evaluation in the creation of a pricing model is a practice suggested by TAUS guidelines. Also, from a more practical point of view, asking more than one linguist to post-edit a whole sample would be too costly. A manual evaluation step allows the collection of observations from a different person than the one involved in the RCT (see Section 3.3) in a relatively short span of time.

Regarding the reliability of such procedure, De Sousa et al. (2011) found a high correlation between subjective scores based on the predicted post-editing effort and the actual post-editing time. On the other hand, works such as the one by Moorkens et al. (2015) have shown that correlations between ratings of predicted post-editing effort and actual temporal effort are only moderate. However, the authors also stated that such re-

sults were probably influenced by the instructions provided to participants. Also, participants were researchers or academic staff members. Results might be different in our scenario, since linguists dealing with translation and post-editing tasks on a daily basis are involved.

Developing a manual evaluation method and an approach to turn the scores assigned by the evaluator into a discount was arguably the most complex part in the creation of the whole pipeline. Due to confidentiality reasons, we cannot reveal the formula currently in use at Acolad. Nevertheless, in the rest of this section details will be provided regarding the manual evaluation setup and method.

A sample of 3,000 words is randomly extracted from a held-out set of sentence pairs and the source text is machine translated with the engine that is being tested. This sample size was chosen to collect a reasonable number of observations, at the same time without asking the evaluator to score a high number of sentences, which might be perceived as a repetitive task, introducing a fatigue effect.

One evaluator is asked to evaluate the sample by assigning a score to each sentence. As in Specia (2009), scores range from 1 to 4 and describe the amount of editing that would be needed in a full post-editing scenario[5]:

1. No editing required
2. Minor editing required. Edits in these sentences are usually related to word order, wrong word class, wrong use of plural/singular forms, etc.
3. Major editing required. Edits require quite some effort, but post-editing is still more efficient than translating from scratch
4. Re-translation required: post-editing would be less efficient than translating from scratch

The evaluator might be negatively influenced by a preconception against MT, or by low quality in the first sentences of the sample, since human evaluation is inherently subjective. To reduce the number of sentences assigned the worst score despite their acceptable quality, we ask the evaluator to provide a correct version of the sentence when this is labelled with the worst score (4). Also, the evaluator is paid based on the time required to finish the task, so as to avoid a situation whereby the outcome of the task is influenced by unfair per-word rates.

At the end of the evaluation, each score is treated as a fuzzy match percentage. For example, sentences with the best score (1) are treated as 100% matches, while sentences with the worst score (4) are treated as 0% matches. Based on these percentages and on the number of words assigned to each score, a formula determines the coefficient in a way similar to the one used to compute weighted word counts in CAT tools.

It is worth noting that a penalty is added to 0% matches, assuming that re-translating an MT output would take longer than translating the same sentence from scratch. The post-editor would need to first read and analyse both the source and the target sentence, and then to delete and re-translate the latter. A 5% margin is then added to the coefficient, in order to take into account the subjectivity of manual scores. Henceforth, the resulting coefficient will be referred to as the *manual evaluation coefficient* (see Section 3.4).

3.3 Real condition test

In the RCT, a sample of 5,000 words is randomly extracted from a held-out set of sentence pairs and machine translated by the engine that is being tested. This sample size was chosen because it should allow the measurement of productivity over a reasonable span of time. The machine translated sample is sent to one post-editor, who is asked to perform full post-editing on the output. The RCT and the manual evaluation (see Section 3.2) are carried out by two different linguists. At this stage, reliable information on the discount to be applied is not available. For this reason, the post-editor taking part in the RCT is paid based on his/her per-word rate for translation from scratch. The post-editor can choose his/her preferred CAT tool to perform this task. This is in line with previous works suggesting that experiments on post-editing productivity should be carried out with the same tools that are normally used in the translation industry (Macken et al., 2020; Läubli et al., 2013).

There are two main outputs of this test. First, the editing effort is obtained computing HTER (Snover et al., 2006) between the raw output and its post-edited version. Time spent on the whole task is tracked by the post-editor, who then provides this datum at delivery. HTER is not used to

[5]Differently from the approach present in this paper, in Specia (2009) the worst score is 1 and the best score is 4.

compute a coefficient, since it might not take into account all edits performed on a text/sentence or the required cognitive effort (Lacruz et al., 2014). Moreover, the number of edits might not correlate with time measurements (Tatsumi, 2009; Macken et al., 2020).

On the other hand, combining HTER with temporal effort provides an overview on productivity and a better understanding of how time was used. The informativeness of a comparison between these two types of post-editing effort will be shown in Section 4.

The time spent on the whole task and the total number of source words are used to compute the result of this test (henceforth, the *RCT coefficient*). First, we retrieve the throughput (source words per hour). Then, we compute the difference between the post-editing throughput and an estimated throughput for translation from scratch, i.e. 357 words per hour (the language industry standard of 2,500 words per day if we consider a seven-hour workday). The ratio between the resulting difference and the estimated throughput for translation from scratch provides the *increased speed*. To conclude, the RCT coefficient is computed using the following formula:

$$RCT\ coeff. = \frac{1}{1 + increased\ speed} \quad (1)$$

Our method is similar to the one used in Plitt and Masselot (2010), and in Guerberof (2014), except for two differences. First, the RCT coefficient measures the throughput improvement, while the formula reported by Plitt and Masselot and by Guerberof (2014) focuses on time savings (i.e. throughput improvement is subtracted from 100). As a result, if the RCT formula outputs a 77% coefficient, the formula by the authors mentioned above would report a 23% of time saved. Also, in Guerberof (2014) and Plitt and Masselot (2010) the actual throughput for translation from scratch is computed, which is not possible for us due to time and budget constraints (this limitation is further discussed in Section 3.4 and 6).

At the end of the task, the post-editor is asked to fill in a feedback form, in which he/she rates the quality of the whole output assigning a score from 1 (worst score) to 5 (best score) to each of the following categories: accuracy, fluency, terminology translation, formatting and punctuation. An optional field for comments is also provided. For cases in which a high number of terminology issues are spotted, post-editors can list examples of wrong target terms found in the output, together with their source and the correct version. This module helps understanding if a low productivity is confirmed by low quality judgements, or if a perceived low quality is contradicted by a high productivity rate. Receiving a general feedback from the final users of our engines can be useful in different ways. As mentioned in Section 1, besides being recommended by TAUS guidelines, this feedback should help post-editors to feel more involved in MT-related processes. On the other hand, our MT team can undoubtedly benefit from linguists' feedback. It is also worth noting that this module is filled in after each post-editing task, so that quality can be consistently monitored thanks to the work of post-editors.

3.4 Computing the final coefficient

If no specific issues were identified during any of the steps described above, the RCT coefficient and the manual evaluation coefficient are averaged to obtain the *final coefficient*. This will be then subtracted from the per-word rate for no-match segments when MT is used.

When averaging the two coefficients, the result is rounded up to the nearest whole number. This is done – together with the margin added to the manual evaluation coefficient and introduced in Section 3.2 – to take into account variability between different post-editors or different texts in terms of productivity. As a matter of fact, rounding up the coefficient reduces the discount. Although the best solution would be to have more than one evaluator and more than one post-editor being involved in the pipeline for each engine test (and a larger sample to be evaluated or post-edited) this is often not feasible due to either budget or time constraints (see Section 6). Only one linguist is usually involved in each task. He/she is chosen from among the linguists who often translate for the customer for which the engine was created. On the other hand, if the outcomes of the test steps are contradictory, too high, or too low, the test is repeated with a different linguist.

From a practical point of view, a further step is required once the coefficient has been computed. Having a large number of engines – with a specific coefficient each – and being a company with a large number of employees, an effort has to

Evaluation step	Outcome	En–De		De–It	
		Fashion	Medical	Medical	Legal
Automatic score	**TER**	61.60	49.86	46.75	48.88
RCT	**HTER**	41.00	29.89	40.77	34.45
	Coefficient	59%	81%	62.00%	72%
Manual evaluation	**Coefficient**	66.00%	80.50%	63.70%	74.20%
Final coefficient		63%	81%	63%	73%

Table 1: Results of each step of our test pipeline for two language combinations (En–De and De–It) on engines developed for customers in the following domains: Fashion, Medical, Legal. For consistency with the coefficients, (H)TER values are presented as a percentage score. TER values (first row) were obtained comparing the MT output to a pre-existing reference translation, while data in the other rows are based on the work by translators involved in the test pipeline. The final coefficient is an average – rounded up – of the other two coefficients and is subtracted from the rate for no match segments. The lower the coefficient, the higher the productivity and the discount.

be made to make sure that project managers can quickly find the best engine for their project (if any) and that they know which discount to apply. For this reason, we developed an automation step that, for each project, suggests the engine that suits the project manager's needs and provides its coefficient. Also, newsletters are sent out to project managers every time a new set of engines is ready to be used.

4 Use cases

4.1 MT engines

After having introduced the different steps of our test pipeline and their outcomes, in this section we present 4 use cases (see Table 1), each of them related to an engine trained for a specific customer. For confidentiality issues, we only reveal the domain the customers operate in. Two engines are trained on English–German data and their domains are fashion and medical. Two engines are trained on German–Italian data and their domains are medical and legal.

The rationale behind the choice of En–De and De–It is that in the first language combination the evaluation steps were carried out on a Germanic language, while in the second language combination, evaluations were carried out on a Romance language. Since different issues might be found in the output, e.g. based on the target language syntax, the post-editing effort (and therefore evaluation results) might be influenced by them. After having chosen the language combinations, medical, legal and fashion were chosen to provide examples for 3 domains with wide differences in terms of sentence structure and terminology.

4.2 Results

Looking at Table 1, we can see that there usually is a large difference between TER in the automatic score step and HTER in the RCT – except for the Medical De–It engine. This is to be expected since the reference test used for TER is a manual translation produced from scratch, while in the case of HTER we are measuring the edit distance between a raw MT output and its post-edited version. At the same time, we are reporting the HTER for the RCT for the sake of completeness and comparison, although – as explained in Section 3.3 – the RCT coefficient is computed based on the productivity increase.

The Fashion En–De engine, which has a TER score of 61.60, seemed to provide rather poor quality. However, once the test pipeline was completed, it was shown to be able to increase productivity by 37%. The exact same result was obtained by the Medical De–It engine. The other Medical engine (En-De) reached a 81% final coefficient – the highest value in the table and thus the worst productivity increment. To conclude, Legal De–It engine reached a 73% final coefficient. According to our coefficients, all engines are thus able to produce an output that increases post-editor productivity by at least 19%.

Comparing the two outcomes of the RCT for all engines, we see many discrepancies. In the fashion En–De engine, for example, 41% of the sample was edited. However, productivity increased by 41% according to the coefficient. In Section 3.3 the low correlation between HTER and productivity measurements was illustrated. This becomes evident when considering RCT results for Medical En–De, Medical De–It and Legal De–It. An HTER increase corresponds to a decrease of the

RCT coefficient, and thus to a productivity increment, confirming that a lower number of edits does not necessarily imply a lower post-editing time and thus a higher productivity.

The RCT coefficient and the manual evaluation coefficient are similar for three engines out of four, with their differences ranging from 0.5 to 2.2%. For fashion En–De, they differ by 7%. Since the manual evaluation coefficient is higher (66%), it is possible that the fashion En–De evaluator was particularly strict when manually scoring sentences, thus causing the coefficient to increase. In the feedback form for the RCT, the post-editor stated that the source text was often translated literally or using an incorrect sentence structure. As shown by Koponen (2012), target sentences involving more reordering tend to be perceived as requiring a high editing effort. It is thus possible that sentences in the manual evaluation sample received a score that reflected a higher post-editing effort than that actually required during RCT.

5 Discussion

We have presented four use cases of our pipeline to test an MT engine and estimate the productivity increase that can be reached when its output is used in post-editing tasks.

Results in Table 1 have shown that when a higher HTER is computed, productivity increases and *vice versa*. Indeed previous works found a low correlation between HTER and temporal effort (see Section 3.3). For example, small edits might require a high cognitive and temporal effort, thus having a large negative impact on productivity.

The small differences between the RCT coefficient and the manual evaluation coefficient seem to suggest that the approach adopted to estimate post-editing effort through manual scoring reflects the actual temporal (and cognitive) effort. This might not always be the case as testified by the fashion En–De engine. Therefore, keeping both coefficients and computing an average of the two seems to be the best solution.

6 Conclusions and future work

The present work aimed at introducing our test pipeline to measure MT quality and engine-specific discounts. Future work might present a higher number of use cases than those described in Section 4.2, in order to provide a better overview of how the pipeline works. Also, adding examples of raw sentences together with the score assigned to them during the manual evaluation and their post-edited versions can better explain differences between the two coefficients or between the two outcomes of the RCT. However, it has to be taken into account that it is often not possible to provide such examples due to confidentiality issues.

The feedback module described in Section 3.3 has been recently added to our pipeline. Before that, post-editors were only asked to provide a general feedback on the output quality and its main issues. However, such general feedback was often too generic and/or difficult to interpret. For this reason, in this paper we are not reporting the feedback received, except for the fashion En–De engine in Section 4.2. Future work could compare HTER and the coefficients with the results of the new feedback module, discussing the accuracy and fluency ratings assigned to the four engines.

Our MT group is constantly working on the improvement of our processes, including improvements of our test pipeline. In the future, language-specific or domain-specific formulas to compute the two coefficients could be developed, since productivity can be highly influenced by the text and/or domain post-editors are working on.

Also, as introduced in Section 3.3, using an average productivity increase to compute discounts to be applied to the work of any post-editor does not take into account differences between professionals. Several studies on post-editing concluded that there is a high degree of variability between post-editors when it comes to productivity (Macken et al., 2020; Guerberof Arenas, 2014; Plitt and Masselot, 2010). However, we fully agree with Guerberof (2014), who argues that using an average productivity increase to compute discounts for all post-editors is not beneficial for less productive post-editors, but further maintains that this is what happens with fuzzy match discounts as well, and finding a solution is not an easy task. To improve the fairness of our discounts, when allowed by deadlines and by the budget, we plan to involve more than one linguist per stage.

To conclude, the authors are aware of the need to develop a pipeline that can be applied to all production tasks involving post-editing. Producing continuous reports on the output quality of an engine – and how this influences productivity – is of the essence for an increasingly efficient and fair

use of MT, similarly to what has been introduced by TAUS Dynamic Quality Framework (DQF).[6] However, developing such a complex pipeline will require a long research and pilot phase.

These possible improvements should not hide the fact that, to the best of our knowledge, this is the first work presenting a company's approach to test engines and method of computing its specific discount rates, reporting also on examples of the practical use of this pipeline.

Acknowledgements

The authors would like to thank the whole Acolad MT team for its contribution to the development of the processes described in the present work, and two anonymous reviewers for insightful comments on the first draft of this paper.

References

Cattelan, Alessandro. 2013. A fair rate for post-editing. https://bit.ly/38FO1Pj. Last accessed: 2020-04-20.

de Sousa, Sheila C. M., Wilker Aziz, and Lucia Specia. 2011. Assessing the post-editing effort for automatic and semi-automatic translations of DVD subtitles. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 97–103, Hissar, Bulgaria, September. Association for Computational Linguistics.

Elia, EMT, EUATC, FIT Europe, GALA, and LIND. 2018. 2018 Language Industry Survey – Expectations and concerns of the European language industry. https://bit.ly/3bogf3X. Last accessed: 2020-04-20.

Elia, EMT, EUATC, FIT Europe, GALA, and LIND. 2019. 2019 Language Industry Survey – Expectations and concerns of the European language industry. https://bit.ly/3btV2pf. Last accessed: 2020-04-20.

Escartín, Carla Parra, Hanna Béchara, and Constantin Orasan. 2017. Questing for quality estimation a user study. *Prague Bull. Math. Linguistics*, 108:343–354.

Guerberof Arenas, Ana. 2014. Correlations between productivity and quality when post-editing in a professional context. *Machine Translation*, 28(3/4):165–186.

Koponen, Maarit. 2012. Comparing human perceptions of post-editing effort with post-editing operations. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 181–190, Montréal, Canada, June. Association for Computational Linguistics.

Krings, Hans P. 2001. *Repairing Texts: Empirical Investigations of Machine Translation Post-Editing Processes*. Kent State University Press, Kent, Ohio.

Lacruz, Isabel, Michael Denkowski, and Alon Lavie. 2014. Cognitive Demand and Cognitive Effort in Post-Editing. In *Proceedings of the Third Workshop on Post-Editing Technology and Practice*, pages 73–84. Association for Machine Translation in the Americas.

Läubli, Samuel, Mark Fishel, Gary Massey, Maureen Ehrensberger-Dow, and Martin Volk. 2013. Assessing post-editing efficiency in a realistic translation environment. In O'Brien, Sharon, Michel Simard, and Lucia Specia, editors, *Proceedings of MT Summit XIV Workshop on Post-editing Technology and Practice (Nice, September 2, 2013)*, pages 83–91, Allschwil, September. European Association for Machine Translation.

Lizuka, Izabella. 2018. Isn't it time to embrace machine translation post-editing? The localization use case for MT. https://bit.ly/3cJs0CF. Last accessed: 2020-04-20.

Macken, Lieve, Daniel Prou, and Arda Tezcan. 2020. Quantifying the effect of machine translation in a high-quality human translation production process. *Informatics*, 7(2).

Moorkens, Joss, Sharon O'Brien, Igor A. L. da Silva, Norma B. de Lima Fonseca, and Fábio Alves. 2015. Correlations of perceived post-editing effort with measurements of actual effort. *Machine Translation*, 29:267–284.

Plitt, Mirko and François Masselot. 2010. A productivity test of statistical machine translation post-editing in a typical localisation context. In *Prague Bulletin of Mathematical Linguistics, 93:7–16, 2010. URL http://ufal.mff.cuni.cz/pbml/93/art-plitt-masselot.pdf*.

Scarton, Carolina, Mikel L. Forcada, Miquel Esplà-Gomis, and Lucia Specia. 2019. Estimating post-editing effort: a study on human judgements, task-based and reference-based metrics of mt quality.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachussets.

Specia, Lucia, Nicola Cancedda, Marc Dymetman, Marco Turchi, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *In EAMT*, pages 28–35.

[6] DQF is a framework to evaluate quality and productivity for translation tasks. https://bit.ly/2VVfVUP

Tatsumi, Midori. 2009. Correlation between automatic evaluation metric scores , post-editing speed , and some other factors. In *Proceedings of the AMTA 2012 Workshop on Post-editing Technology and Practice*, pages 69–77. Association for Machine Translation.