Post-editing of Machine Translation

Post-editing of Machine Translation: Processes and Applications

Edited by

Sharon O'Brien, Laura Winther Balling,
Michael Carl, Michel Simard and Lucia Specia

CAMBRIDGE SCHOLARS
PUBLISHING

Post-editing of Machine Translation: Processes and Applications,
Edited by Sharon O'Brien, Laura Winther Balling, Michael Carl,
Michel Simard and Lucia Specia

This book first published 2014

Cambridge Scholars Publishing

12 Back Chapman Street, Newcastle upon Tyne, NE6 2XX, UK

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN (10): 1-4438-5476-X, ISBN (13): 978-1-4438-5476-4

TABLE OF CONTENTS

FOREWORD

Post-editing is possibly the oldest form of human-machine cooperation for translation, having been a common practice for just about as long as operational machine translation systems have existed. Recently however, there has been a surge of interest in post-editing among the wider user community, partly due to the increasing quality of machine translation output, but also to the availability of free, high-quality software for both machine translation and post-editing.

Technology and the challenges of integrating post-editing software and processes into a traditional translation workflow are at the core of research in post-editing. However, this topic involves many other important factors, such as studies on productivity gains, cognitive effort, pricing models, training and quality. This volume aims at covering many of these aspects by bringing together accounts from researchers, developers and practitioners on the topic. These are a compilation of invited chapters from work presented at two recent events on post-editing:

1. The first Workshop on Post-editing Technology and Practice (WPTP), organised by Sharon O'Brien (DCU/CNGL), Michel Simard (CNRC) and Lucia Specia (University of Sheffield) and held in conjunction with the AMTA Conference in San Diego, October 28, 2012; and
2. The International Workshop on Expertise in Translation and Post-editing Research and Application (ETP), organised by Michael Carl, Laura Winther Balling and Arnt Lykke Jakobsen from the Center for Research and Innovation in Translation and Translation Technology and held at the Copenhagen Business School, August 17-18, 2012.

The goals of the two workshops were different, and so was their format. ETP[1] had two related purposes: The first was to explore the process of post-editing machine translation compared with from-scratch translation, and the role of expertise in both processes. The second was to discuss to what extent knowledge of the processes involved in human translation and post-editing might shape advanced machine translation and computer-

assisted translation technologies. It invited short summaries to be submitted, with oral presentation slots given to all participants with accepted summaries.

WPTP[2], on the other hand, issued an open call for papers to be published in the workshop proceedings and presented either orally or as posters, and offered slots for post-editing software demonstrations. It focused on research assessing the weaknesses and strengths of existing technology to measure post-editing effectiveness, establish better practices, and propose tools and technological PE solutions that are built around the real needs of users. Despite the wide range of topics in both workshops, most of the actual work submitted and presented at ETP concentrated on studies of the post-editing process, while work at the WPTP workshop focused on technology for post-editing and their impact on productivity.

This volume aims at bringing these two perspectives together in one book. It compiles contributions of 28 authors into 13 chapters, which are structured in three parts: (I) macrolevel processes, (II) microlevel processes and (III) guidelines and evaluation. We hope that this compilation will contribute to the discussion of the various aspects involving post-editing processes and applications and lead to a better understanding of its technological and cognitive challenges. Finally, we would like to thank all authors and reviewers for their committed work

The editors

Notes

[1] http://bridge.cbs.dk/platform/?q=ETP2012
[2] https://sites.google.com/site/wptp2012/

INTRODUCTION

MIKE DILLINGER

These are very exciting times for translation research

As global communication and commerce increase, the importance and scale of translation have skyrocketed. As technology becomes more complex and competition leads to accelerating innovation, exponentially more content has to be translated not only much more quickly but also much more cheaply than ever before. Consequently, it has become crucial to understand how to make the translation process as quick, accurate, and effective as possible – both with and without software tools. In this context, the role of machine translation and post-editing MT output have taken on new importance.

In an equally significant shift, translation researchers have shifted away from studies of conceptual and pedagogical issues to a new focus on systematic empirical data about real-world translation tasks – data about industrial and cognitive translation *processes*. As a result, there are more researchers, more numerous and more sophisticated tools for research, and more and more detailed data than were available only ten years ago.

Where is translation research going?

Translation research is quickly moving toward building detailed process models. These are step-by-step descriptions of exactly what happens in individual translators as they translate source texts or post-edit source text/draft translation pairs. For each step, we will soon be able to identify the text, task, and translator characteristics that have the biggest impact on performance. As we generalise across translators and texts, we can identify optimal practices – based on reliable data rather than only on intuition – that will have a significant impact on the translation industry.

What would a processing model of post-editing look like?

It would start with a framework of steps that make up text comprehension in L1 and L2. We know already that monolingual text comprehension plays a key role in post-editing. Fortunately, both theory and research in this field are very rich and detailed. However, post-editing raises new questions for research. For example, do the post-editor's comprehension strategies change when reading about an unfamiliar topic specifically for post-editing or for translation? Do the specific characteristics of MT output change reading strategies or performance significantly? Do post-editors need more or different topic or linguistic knowledge than readers do? Recall that one common use case for post-editing MT deals with technical information that most translators are not very familiar with. Comprehension clearly varies based on source-text characteristics, as well as on the post-editor's language skills and topic knowledge. Future studies will measure post-editors' comprehension in L1 and L2 more directly and explore which source-text characteristics affect which steps of the post-editing process.

Another step (and a defining core competence) of post-editing and translation is the ability to judge the equivalence of two sentences in different languages after they have been understood. However, there is limited research even in how monolinguals detect similarities and differences across sentences in the same language (the vast research into how people perceive similarities and differences of *words* seems not to have continued with sentences). Which sentence characteristics or typological differences make it easier or harder to judge equivalence across languages? Do post-editors pay more (or less) attention to some sentence characteristics than do translators? Post-editors also have to switch often between L1 reading and L2 reading – does this switch slow them down or affect accuracy? The research literature on monolingual revising is definitely a good place to start, at the very least as a detailed process model to start from. Judging equivalence across languages seems to be a new area of study and may become a defining area of translation research.

In yet another step, post-editors have to produce sentences and texts – or edit existing options. Again, there is a rich existing research literature on sentence production – not as well developed as the comprehension literature, but it focuses on normal, monolingual writing tasks that usually start from conceptual plans rather than from other texts. Are the production processes during post-editing (or during translation) different from normal, monolingual writing-from-ideas? *How* are they different? Do

the post-editors' *writing* skills in L1 and L2 affect how (and how well) this happens? Is post-editing easier or harder than monolingual revising, and why? Are some kinds of edits easier or harder than others, and why?

One likely possibility is that both text comprehension and text production will be very similar in monolingual tasks and in bilingual tasks such as post-editing and translation. The key novelty – and crucial difference – for bilingual tasks, then, may turn out to be the ability to compare sentences (and texts) across languages, in terms of both literal meaning and the culturally determined patterns of inference and connotation that different phrasings will entail. Moving forward, translation researchers will check these possibilities much more carefully then identify and focus on the abilities that make post-editing and translation so special.

This discussion shows that there are many factors to consider each time we study post-editing. Too many factors, in fact. Methodologically, we have three basic ways to deal with the factors that we know about: *ignore* the influence of these factors, *control* the effects of these factors, or *focus* on their influence. Standard experimental practice is to focus on a couple of factors, control the effects of as many known factors as practically possible, and ignore the rest – then change them in subsequent studies. To provide more detailed results, future studies will control more and more relevant factors.

Where is the field now?

The present volume shows that the study of translation processes is full of promise – there is much more to come. There is a clear emphasis in these chapters on developing and testing the wide range of methods, tools, and datasets that we need to start building the kinds of process models sketched above. There are great examples of how to apply sophisticated statistical methods to post-editing data, such as principal components analysis and multiple regression. There are exciting new tools for collecting (and integrating) data about keystrokes, eye movements, and pauses as post-editors work in real time. There are reports on growing and increasingly detailed datasets that have been built with these tools (and with others) – and that can be analysed in very many different ways.

Note that the studies in this volume are all very difficult to do because they require skills and detailed understanding of concepts from multiple disciplines: translation, linguistics, cognitive psychology, applied statistics, process engineering, management, software engineering, computational linguistics, and many others. Since there are very, very few researchers today with all of this background, interdisciplinary collaboration is

essential. For the reader, this means that each chapter will have a surprising and different mix of interdisciplinary perspectives, methods, and data.

Unavoidably, in beginning stages of interdisciplinary research, there are methodological errors. Don't let them distract you from the fact that the questions that these studies pose and the tools and datasets that they have succeeded in building constitute significant progress and a sign of more progress to come – even in the cases where the analyses are weak and the conclusions are not so reliable. This is normal for new areas of research – it simply reinforces the need and opportunity for intense interdisciplinary collaboration.

The contributions to this volume seem to fall naturally into three parts: (I) studies of macro-level translation processes, (II) studies of micro-level translation processes and (III) theoretical studies.

Studies of macro-level translation processes

These studies focus on the industrial translation process from receiving the client's source text to delivering the client's target text. In these studies, the individual translator plays a crucial role but is not the focus of research. Instead, the chapters seek to establish reliable baselines for the whole translation process, with and without the introduction of specific tools, training, management techniques, etc. They generally focus on overall, after-the-fact measures such as productivity or speed. The time frame for these processes is days or weeks.

1. **Zhechev** describes in detail how productivity in very mature post-editing processes varies across language pairs and across source documents for different products.
2. **Silva** insightfully describes how rolling out new post-editing processes can affect a translation company as a whole and provides valuable lessons learned.
3. **Guerberof** focuses on how different translator characteristics may affect overall productivity.

Studies of micro-level translation processes

These studies focus on the individual translator's behaviours, preferences, and cognitive processes – often monitoring the translator in near-real time by measuring eye movements, keystrokes, pauses, etc. as the translator is working. These chapters seek to establish reliable information about how

and how much a wide range of factors affects the individual translator during the translation task itself. The time frame for these processes is milliseconds or seconds.

4. **Depraetere, De Sutter & Tezcan** measure post-editing effort as the similarity between MT output and the final post-edited translation and find that (i) MT enhances the translator's productivity, even if translators are in the initial stages of their careers, (ii) MT does not have a negative impact on the quality of the final translation, and (iii) post-editing distance is more stable across informants than are human evaluation scores, so distance is a potentially more objective measure.
5. **Teixeira** explores the hypothesis that translation metadata might be useful for translators. While too many translation options would be a time drain in hectic localisation projects, the GUI should account for personalisation/customisation, so that it can be adapted to different work styles.
6. **Moran, Lewis & Saam** describe an exciting new tool (iOmegaT) for collecting detailed online data in an ecologically valid translation environment – and some preliminary data gathered with it. They enhanced an open-source translation environment – that is very similar to the industry-standard Trados environment – with a range of logging and reporting functions. Their detailed measurements suggest that post-editing is about twice as fast as translating from scratch (across several languages, with similar content) and they alert us to the fact that translators often go back and review their translations so measures of first-pass translation speed may be misleading.
7. **Elming, Winther Balling & Carl** describe the CASMACAT workbench in detail and show how useful expertly done regression analysis can be with a first dataset that they collected. They showed that post-editing keystroke ratio is a better predictor of post-editing time divided by translation time than edit distance is.
8. **Aziz, Koponen, & Specia** show very clearly how detailed attention to source-text characteristics, sub-sentence post-editing time, and a fruitful mix of qualitative and quantitative analysis lead to insightful and precise results. This is an interesting example of one effective way to use the Principal Components Analysis. Careful readers will notice that they generalised about different kinds of post-editing units because there was not enough data to generalise about translator similarities or differences.

9. **Čulo, Guermuth, Hasen-Schirra & Nitzke** give interesting examples of qualitative differences in strategies that are used to edit, post-edit, and translate the same texts, extracted from a new multilingual dataset built using the CASMACAT workbench. Their key idea is to compare post-editing with both monolingual revising and with translation, so we can be sure that further generalisations from their analyses will provide unique insights about how these processes compare.
10. **Mesa-Lao** correlated source-text complexity with an interesting range of on-line measures during both translation and post-editing tasks. His attention to the details of the source texts means that as more of this kind of data becomes available, it will be possible to make more detailed generalisations about the effects of the source text.
11. **Lacruz & Shreve** focus on patterns of pausing during post-editing, extending early studies of selective attention during shadowing and interpreting done by Anne Treisman in the 1970s. Their finding that more cognitive effort seems to be associated with fewer pauses raises interesting questions when compared to earlier research that concluded the opposite.

Theoretical studies

These studies step back from detailed data to identify the concepts that we need to understand in more detail.

12. **Melby, Fields & Housely** provide detailed specifications for describing post-editing tasks by specifying all of the relevant parameters of this kind of translation job, including different notions of translation quality. They make the very important point that studies of translation processes will lead to inconsistent results if researchers do not define and measure the quality of the output translation in explicit and similar ways.
13. **Rico & Ariano** define detailed and insightful guidelines for post-editing based on their experience rolling out new post-editing processes at a company.

These three types of studies are all equally necessary for the progress of the field. Studying macro-level translation processes provides context, relevance, and crucial practical motivation for the other two types of studies. Without the link to economic consequences that these macro-level

studies contribute, the other studies run the risk of becoming academic exercises that are ignored in practice. Studying micro-level translation processes adds support and more detailed understanding to macro-level studies and suggests directions for specific improvements in practice. These micro-level studies explain just why (and in more detail), for example, some tools or procedures work better than others do in a macro-level setting. In addition, theoretical studies keep everyone honest by checking key concepts in detail and integrating results to check for consistency – so that everyone's results are more reliable.

Mike Dillinger
California, USA
June, 2013

PART I:

MACRO-LEVEL TRANSLATION PROCESSES

CHAPTER ONE

ANALYSING THE POST-EDITING OF MACHINE TRANSLATION AT AUTODESK

VENTSISLAV ZHECHEV

Abstract

In this chapter, we provide a quick overview of the machine translation (MT) infrastructure at Autodesk, a company with a very broad range of software products with worldwide distribution. MT is used to facilitate the localisation of software documentation and UI strings from English into thirteen languages. We present a detailed analysis of the post-edited data generated during regular localisation production. Relying on our own edit-distance-based JFS metric (Joint Fuzzy Score), we show that our MT systems perform consistently across the bulk of the data that we localise and that there is an inherent order of language difficulty for translating from English. The languages in the Romance group typically have JFS scores in the 60–80% range, the languages in the Slavic group and German typically have JFS scores in the 50–70% range and Asian languages exhibit scores in the 45–65% range, with some outlying language/product combinations.

Introduction

Autodesk is a company with a very broad range of software products that are distributed worldwide. The high-quality localisation of these products is a major part of our commitment to a great user experience for all our clients. The translation of software documentation and user interface (UI) strings plays a central role in our localisation process and we need to provide a fast turnaround of very large volumes of data. To accomplish this, we use an array of tools—from document- and localisation-management systems to machine translation (MT).

In this chapter, we focus on the detailed analysis of the post-editing of MT during the localisation process. After a quick look at our MT infrastructure, we focus on the productivity test we organised to evaluate the potential benefit of our MT engines to translators. We then turn to the analysis of our current production post-editing data.

MT Infrastructure at Autodesk

In this section, we briefly present the MT infrastructure that we have built to support the localisation effort at Autodesk. For an in-depth discussion, see Zhechev (2012).

We actively employ MT as a productivity tool and we are constantly improving our toolkit to widen our language coverage and achieve higher quality. At the core of this toolkit are the tools developed and distributed with the open-source Moses project (Koehn et al. 2007). Currently, we use MT for translating from US English into twelve languages: Czech, German, Spanish, French, Italian, Japanese, Korean, Polish, Brazilian Portuguese, Russian, Simplified and Traditional Chinese (hereafter, we will use standard short language codes). We recently introduced MT for translating into Hungarian in a pilot project.

Training Data

Of course, no statistical MT training is possible unless a sufficient amount of high-quality parallel data is available. In our case, we create the parallel corpora for training by aggregating data from four internal sources. The smallest sources by far consist of translation memories (TMs) used for the localisation of marketing materials and educational materials. The next source is our repositories for translated User Interface (UI) strings. This data contains many short sentences and partial phrases, as well as some strings that contain UI variables and/or UI-specific formatting. The biggest source of parallel data is our main TMs used for the localisation of the software documentation for all our products.

To ensure broader lexical coverage, as well as to reduce the administrative load, we do not divide the parallel data according to product or domain. Instead, we combine all available data for each language and use them as one single corpus per language. The sizes of the corpora are shown on Figure 1-1, with the average number of tokens in the English source being approximately 13.

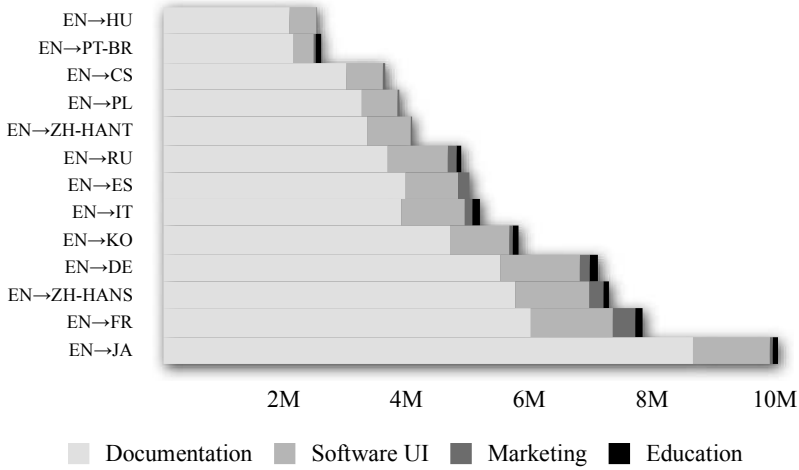

Figure 1-1: Training Corpora Sizes in Millions of Segments

As Figure 1-1 shows, we have the least amount of data for PT-BR and HU, while our biggest corpus by far is for JA. The reader can refer to this chart when we discuss the evaluation of MT performance—it turns out that the difficulty of translating into a particular language from English is a stronger factor there than training data volume.

After we have gathered all available data from the different sources, we are ready to train our MT systems. For this, we have created a dedicated script that handles the complete training workflow. In effect, we simply need to point the script to the corpus for a particular language and—after a certain amount of time—we get a ready-to-deploy MT system. Further information on the training infrastructure can be found in Zhechev (2012).

MT Info Service

We now turn to the MT Info Service that is the centrepiece of our MT infrastructure, handling all MT requests from within Autodesk. This service and all its components are entirely based on the Perl programming language and handle service requests over internal and external network connections over TCP (Transmission Control Protocol).

The first elements of this infrastructure are the MT servers that provide the interface to the available MT engines running in a data centre. At launch time, the server code initiates the Moses translation process. The MT servers receive translation requests for individual segments of text

(typically sentences) and output translations as soon as they are available. For each language that we use in production, we currently have up to seven MT engines running simultaneously on different servers to provide higher overall throughput.

The MT Info Service itself acts as a central dispatcher and hides the details of the MT servers' setup, number and location from the clients. It is the single entry point for all MT-related queries, be it requests for translation, for information on the server setup or administrative functions. It has real-time data on the availability of MT servers for all supported languages and performs load balancing for all incoming translation requests to best utilise the available resources. In real-life production, we often see twenty or more concurrent requests for translation that need to be handled by the system—some of them for translation into the same language. We have devised a simple and easy-to-use API that clients can use for communication with the MT Info Service.

Over the course of a year, the MT Info Service may receive over 180,000 translation requests that are split into more than 700,000 jobs for load balancing. These requests include over one million documentation segments and a large volume of UI strings.

Integrating MT in the Localisation Workflow

Once we have our MT infrastructure in place and we have trained all MT engines, we need to make this service available within our localisation workflow so that raw data is machine translated and the output reaches the translators in due course. We use two main localisation tools—SDL Passolo for UI content and SDL WorldServer for localisation of documentation.

Unfortunately, the current version of Passolo that we use does not provide good integration with MT and requires a number of manual steps. First, the data needs to be exported into "Passolo bundles". These are then processed with in-house Python scripts that send any data that has not been matched against previous translations to the MT info service. The processed bundles are then passed on to the translators for post-editing. Due to limitations of Passolo, the MT output is not visibly marked as such and Passolo has no way to distinguish it from human-produced data. We expect this to be addressed in an upcoming version of the tool.

It is much easier to integrate MT within WorldServer. As this is a Java-based tool, it allows us to build Java-based plugins that provide additional functionality. In particular, we have developed an MT adapter for WorldServer that communicates directly with the MT Info Service over

TCP and sends all appropriate segments for machine translation. The MT output is then clearly marked for the convenience of the translators both in the on-line workbench provided by WorldServer and in the files used to transfer data from WorldServer to standalone desktop CAT tools.

WorldServer presents us with its own specific issues to handle, for a discussion of which we would like to refer the reader to Zhechev (2012).

Product-Specific Terminology Processing

To support the spectrum of domains represented by our broad product portfolio, we needed an effective system that would select product-appropriate terminology during machine translation, as terminology lookup is one of the most time consuming and cognitively intense tasks translators have to deal with. This is particularly true for the data typically found in our software manuals—rich in industry-specific terminology from architecture, civil engineering, manufacturing and other domains.

One solution to this problem would be to create product and/or domain specific MT engines that should produce domain-specific output. Unfortunately, as can be seen in Figure 1-14 below, most of the localisation volume is concentrated in a few flagship products, while the rest of the products have fairly low amounts of data. Trying to train MT engines only on product-specific data is thus destined to fail, as out of the approximately 45 products that we currently localise, only about five have sufficient amounts of TM data for training an operational MT engine.

We could, of course, always train on the whole set of data for each language and only perform tuning and/or language model domain adaptation for each specific product/domain group. However, this would result in as much as 585 different product specific engines (13 languages times 45 products) that need to be maintained, with each further language we decide to localise into adding another 45 engines. The engine maintenance would include regular retraining and deployment, as well as the necessary processing power to have that number of engines (plus enough copies for load-balancing) available around the clock—the latter being particularly important as the software industry moves to agile continuous development of software products, rather than yearly (or similar) release cycles.

Our solution allows us to only train one MT engine per target language and use built-in Moses functionality to fix the product-specific terminology during a pre-processing step. As part of our regular localisation process, product-specific glossaries are manually created and maintained for use by human translators. When new data is sent to the MT Info Service for

processing, the MT request includes the corresponding product name. This allows the selection of the proper product-specific glossary and annotating any terms found in the source data with XML tags providing the proper translations. Moses is then instructed to only use these translations when processing the data, thus ensuring that the MT output has the proper target-language terminology for the specified product.

One drawback of this approach is that the product glossaries only contain one translation per language per term, which is one particular morphological form. This means that for morphologically rich languages like Czech, the product-specific terminology will often carry the wrong morphological form. However, we estimate that the time needed to fix the morphology of a term is significantly less than the time needed to consult the glossaries in the appropriate tools to make sure the source terms are translated correctly.

Our approach also allows us to eschew the tuning step during MT training. Given our broad product portfolio, selecting a representative tuning set is particularly hard and necessarily biases the MT system towards some products at the cost of others. Considering these factors, as well as the level of performance of our non-tuned MT engines, we have decided to bypass the tuning step. We thus save computing time and resources, without losing too much in MT quality.

So far we had a look at the complex MT infrastructure at Autodesk. The question that arises is if there is any practical benefit to the use of MT for localisation and how to measure this potential benefit. We present our answer in the next sections.

Post-Editing Productivity Test

We now turn to the setup of our last productivity test and analyse the data that we collected. The main purpose of the productivity test was to measure the productivity increase (or decrease) when translators are presented with raw MT output for post-editing, rather than translating from scratch.

We are presenting here the results of the third productivity test that Autodesk has performed. The results of the first test in 2009 are discussed in Plitt and Masselot (2010). Each of the tests has had a specific practical goal in mind. With the first productivity test we simply needed a clear indicator that would help us decide whether to use MT in production or not and it only included DE, ES, FR and IT. The second test focused on a different set of languages, for which we planned to introduce MT into production, like RU and ZH-HANS.

The goal of the productivity test described in this chapter was mainly to confirm our findings from the previous tests, to help us pick among several MT options for some languages and compare MT performance across products. In the following discussion we will only concentrate on the overall outcome of the productivity test and on our analysis of the post-editing performance against automatic, edit-distance-based indicators.

Test Setup

The main challenge for the setup of the productivity test is the data preparation. It is obviously not possible for the same translator to first translate a text from scratch and then post-edit an MT version without any bias—the second time around the text will be too familiar and this will skew the productivity evaluation. Instead, we need to prepare data sets that are similar enough, but not exactly the same, while at the same time taking into account that the translators cannot translate as much text from scratch as they can post-edit—as our experience from previous productivity tests has shown. This is further exacerbated by the fact that we need to find data that has not been processed yet during the production cycle and has not yet been included in the training data for the MT engines.

Due to resource restrictions, we only tested nine out of the twelve production languages: DE, ES, FR, IT, JA, KO, PL, PT-BR and ZH-HANS. For each language, we enrolled four translators—one each from our usual localisation vendors—for two business days, i.e. sixteen working hours. We let our vendors select the translators as per their usual process.

We put together test sets with data from four different products, but most translators only managed to process meaningful amounts of data from two products, as they ran out of time due to various reasons (connectivity issues; picked the wrong data set; etc.). These included three tutorials for AutoCAD users and a user's manual for PhysX (a plug-in for 3ds Max). In all cases about one-third of the data was provided without MT translations—for translation from scratch—while the other two-thirds were presented for post-editing MT. The number of segments the individual translators processed differed significantly based on the productivity of the individual translators. The total number of post-edited MT segments per language is shown below in Figure 1-3.

The translators used a purpose-built online post-editing workbench that we developed in-house. While this workbench lacked a number of features common in traditional CAT tools (e.g. TM and terminology search), it allowed us to calculate the time the translators took to look at and translate/post-edit each individual segment. For future productivity tests

we plan to move away from this tool and use, for example, a modified version of Pootle (translate.sourceforge.net) instead, as it is easier to manage and provides typical CAT functionality, or one of the many tools that have been released recently to address this type of testing.

Evaluating Productivity

After gathering the raw productivity data, we automatically removed any outlier segments, for which the translators took unreasonably long time to translate or post-edit. To calculate the average productivity increase resulting from the provision of MT output to translators for post-editing, we needed a baseline metric that would reflect the translator productivity when translating from scratch. Selecting this baseline was a complex task for a number of reasons. We could not have a direct measurement of productivity increase for each individual segment, as translators were not post-editing the same segments they had translated from scratch. Furthermore, the variability in productivity between the different translators for one language, as well as in the individual translator productivity for different products, precluded us from establishing a unified (language-specific) productivity baseline. Instead, we set up separate mean-productivity baselines for each translator-product combination (measured in words per eight-hour business day—WPD), also treating documentation and UI content for the same product as separate sets.

The post-editing productivity for each individual segment within each set was then compared to the corresponding baseline to establish the observed productivity increase (or decrease). The calculated average productivity increase per language is presented in Figure 1-2.

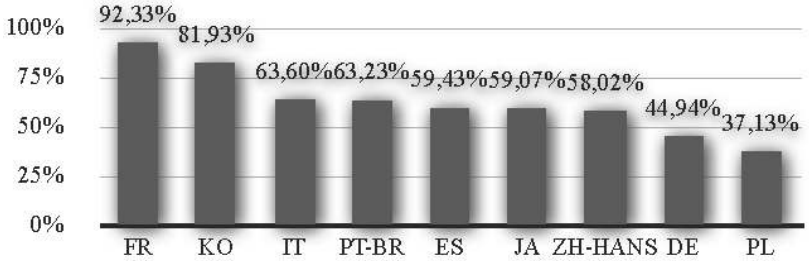

Figure 1-2: Average Productivity Increase when Post-Editing, per Language

A caveat is in order here. Due to the setup of our online workbench, we chose to exclude from the productivity calculation certain translator tasks that are independent of the quality of MT. This includes in particular the

time that translators would usually spend looking up terminology and consulting the relevant style guides. The calculation also does not include any pauses taken for rest, coffee, etc.

Analysing the Post-editing Performance

Going deeper, we went on to analyse the post-edited data using a battery of metrics. The metric scores were computed on a per-segment basis so that we could look for a correlation between the amount of post-editing undertaken by the translators and their productivity increase. The goal of this endeavour was to single out a metric (or several metrics) that we could use for the analysis of our production data, where productivity measurements are not available. This would give us tools to quickly diagnose potential issues with our MT pipeline, as well as to rapidly test the viability of potential improvements or new developments without having to resort to full-blown productivity tests.

The metrics we used were the following: METEOR (Banerjee and Lavie 2005) treating punctuation as regular tokens, GTM (Turian, Shen, and Melamed 2003) with exponent set to three, TER (Snover et al. 2006), PER (Position-independent Error Rate—Tillmann et al. 1997) calculated as the inverse of the token-based F-measure, SCFS (Character-based Fuzzy Score, taking whitespace into account), and WFS (Word-based Fuzzy Score, on tokenised content). The Fuzzy Scores are calculated as the inverse of the Levenshtein edit distance (Levenshtein 1965) weighted by the token or character count of the longer segment. They produce similar, but not equal, results to the Fuzzy Match scores familiar from the standard CAT tools. All score calculations took character case into account. *SLength* denotes the number of tokens in the source string after tokenisation, while *TLength* denotes the number of tokens in the MT output after tokenisation.

After calculating the scores for all relevant segments, we obtained an extensive data set that we used to evaluate the correlation between the listed metrics and the measured productivity increase. The correlation calculation was performed for each language individually, as well as combining the data for all languages. We used Spearman's ρ (Spearman 1907) and Kendall's τ (Kendall 1938) as the correlation measures. The results are shown in Table 1-1.

	Productivity Increase	
	ρ	τ
JFS	0,609	0,439
SCFS	0,583	0,416
WFS	0,603	0,436
METEOR	0,541	0,386
GTM	0,577	0,406
TER	-0,594	-0,427
PER	-0,578	-0,415
SLength	-0,128	-0,087
TLength	-0,143	-0,097

Table 1-1: Correlation of Automatic Metrics to Translator Productivity Increase

We see that among the metrics listed above, WFS exhibits the highest correlation with the measured productivity increase, while METEOR shows the least correlation. The results also show that there is no significant correlation between the productivity increase and the length of the source or translation (cf. the *SLength* and *TLength* metrics). This suggests, for example, that a segment-length-based payment model for MT (e.g. adjusting the MT discount based on segment length) may not be a fair option. Also, we do not need to impose strong guidelines for segment length to the technical writers.

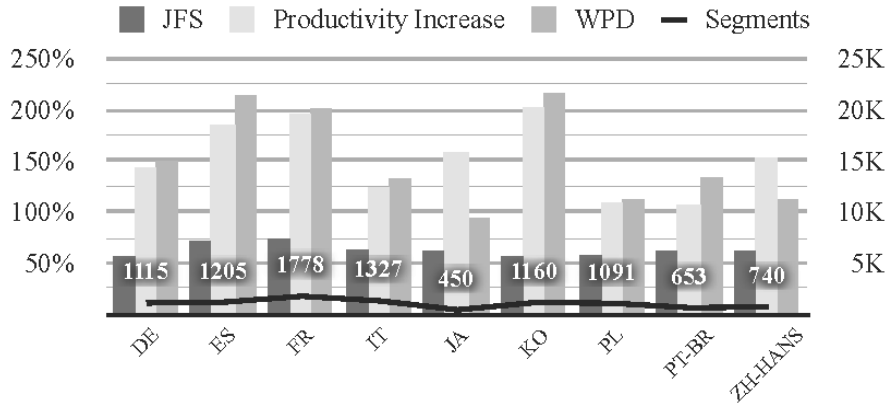

Figure 1-3: Edit Distance and Productivity Data for All Languages

Considering the results, we decided to look for a possibility to create a joint metric that might exhibit an even higher level of correlation. The best available combination turned out to be taking the minimum of SCFS and WFS, which we list in the table as JFS (Joint Fuzzy Score). We also tested using the maximum of SCFS and WFS, as well as other combinations of metrics and different types of means (arithmetic, geometric, etc.). The JFS metric has also an intuitive meaning in that it represents the worst-case editing scenario based on the character and token levels. All other metric combinations we evaluated resulted in lower correlation than WFS. Figure 1-3 presents the JFS scores per language and the corresponding average productivity increase and post-editing speed. It also lists the total number of segments that were post-edited for each language.

In Figures 1-.4–1-11, we investigate the distribution of the JFS scores for the different languages tested. The per-segment data is distributed into categories based on the percentile rank. Due to their particular makeup, we separate the segments that received a score of 0% (worst translations) and those that received a score of 100% (perfect translations) from the rest. For each rank, we show the maximum observed JFS (on the right scale). This gives us the maximum JFS up to which the observed average productivity increase is marked by the lower line on the chart (on the left scale). For all languages, we can observe a sharp rise in the productivity increase for the perfect translations, while otherwise the productivity increase grows mostly monotonically.

Additionally, for each percentile rank, the left bar on the graph shows the percentage of the total number of tokens, while the right bar shows the percentage of the total number of segments.

We do not include a chart for KO, as it does not appear to follow the monotonicity trend and, indeed, our evaluation of the KO data on its own showed a ρ coefficient of only 0,361 for JFS. We suspect that this is due to one of the KO translators ignoring the MT suggestions and translating everything from scratch. Because of this peculiarity of the KO data, we excluded it when calculating the overall results shown in Table 1-1. This also suggests that the productivity increase for KO shown in Figure 1-2 might not be realistic.

It can be argued that we should nonetheless include the KO data in our evaluation, as it represents the real-life scenario of translators being averse to the use of MT. The current trend, however, is for a rise in the level of acceptance of MT, so we expect a decrease in the translator proclivity for ignoring the provided MT output and translating from scratch. Our goal in this test was to discover and analyse the operating parameters of our infrastructure for the case where the MT output is indeed used by the translators.

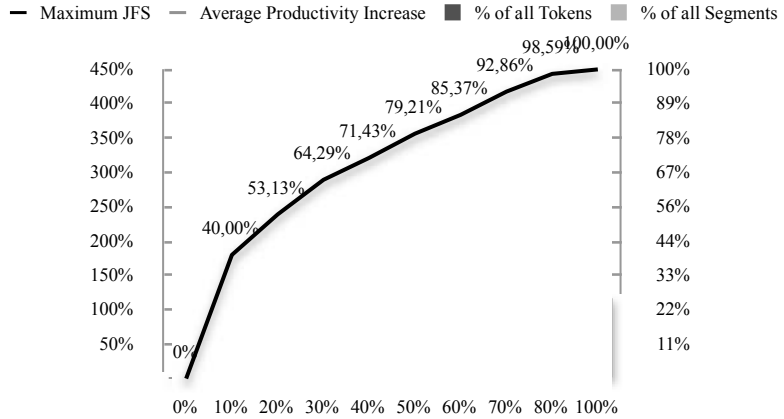

Figure 1-4: JFS to Productivity Correlation FR

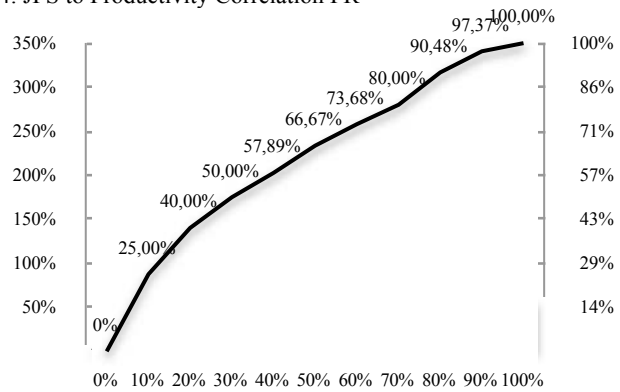

Figure 1-5: JFS to Productivity Correlation IT

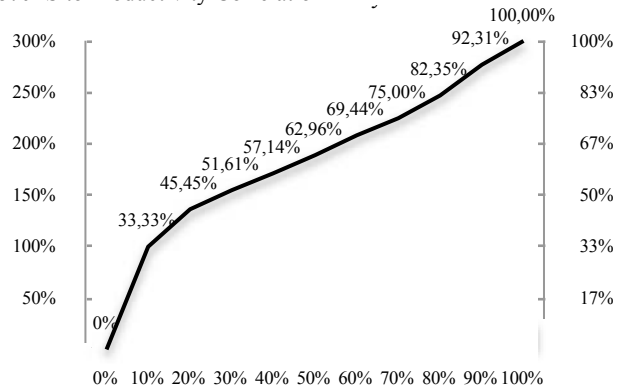

Figure 1-6: JFS to Productivity Correlation PT-BR

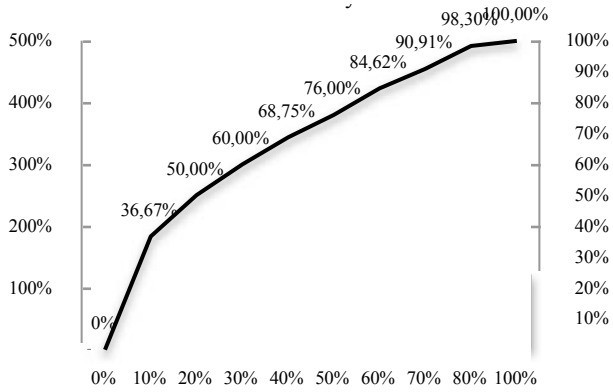

Figure 1-7: JFS to Productivity Correlation ES

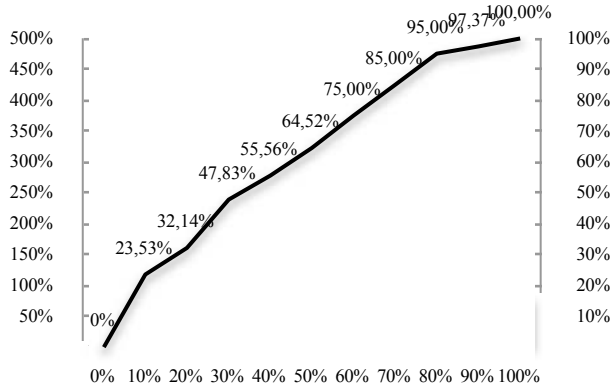

Figure 1-8: JFS to Productivity Correlation JA

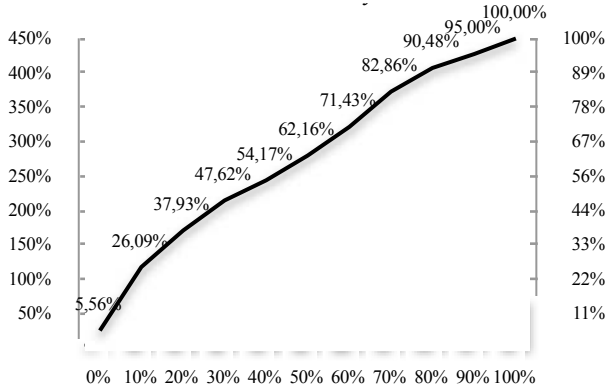

Figure 1-9: JFS to Productivity Correlation ZH-HANS