2012 AMTA 20 Years

The Tenth Biennial Conference of the
Association for Machine Translation in the Americas

Post-Editing Technology and Practice

Sharon O'Brien
CNGL / DCU

Michel Simard
National Research
Council Canada

Lucia Specia
University of Sheffield

SAN DIEGO, CA
OCTOBER 28– NOVEMBER 1, 2012

Proceedings of

WPTP 2012

AMTA 2012 Workshop on

Post-Editing Technology and Practice

Sharon O'Brien, Michel Simard and Lucia Specia
(editors)

28 October 2012
San Diego, California, USA

Foreword

Post-Editing has been around for just about as long as operational machine translation (MT) systems; as such, it is possibly the oldest form of human-machine cooperation for translation. Recently, however, there seems to be a surge of interest for post-editing among the wider user community, partly due to the increasing quality of MT output, but also to the availability of free, high-quality MT software.

Yet, the success of a post-editing operation depends on more than just software, and for every post-editing success story, probably many more failures go unreported. This workshop is an opportunity for post-editing researchers and practitioners to get together and openly discuss the weaknesses and strengths of existing technology, to properly and objectively assess post-editing effectiveness, to establish better practices, and propose tools and technological post-editing solutions that are built around the real needs of users.

The program consists of a mix of oral presentations, posters and software demonstrations. It is a snapshot of the wide variety of scientific and technological work currently taking place.

A number of researchers are tackling the difficult task of understanding the post-editing process itself, for example by studying the relationship between cognitive effort and post-editing time (Koponen et al.), or the relationship between cognitive effort and pauses (Lacruz et al.); others are examining the potential of crowdsourcing post-editing (Tatsumi et al.).

For these sorts of investigation to be effectively carried out, tools are required, specifically those designed for the purpose of observing post-editors and evaluating their work. This workshop features demonstrations and presentations of many such tools: the CASMACAT Workbench (Elming and Bonk), Transcenter (Denkowski and Lavie), PET (Aziz and Specia), and Ruqual (Melby et al.). New technology beyond tools for post-editing *per se* is also taking shape: tools for detecting MT errors (Valotkaite and Asadullah), tools for correcting them (Mundt et al.), or complete online post-editing frameworks with integrated MT functionalities (Penkale and Way).

Post-editing experiments are complex and costly, and it is critical that the experimental evidence that results is preserved and shared between researchers. This is the motivation behind the CRITT TPR database (Carl).

Finally, a special session on *Post-editing experiments in operational settings* will feature accounts of "real-life" experiments, such as recently took place at Autodesk (Zhechev; Beregovaya and Moran) and various EU institutions (Poulis and Kolovratnik), as well as a report on GALA's ongoing "Post-editing Experiment" (Canek).

We wish to thank the AMTA people for making this event possible, providing logistical and moral support at all times. We must also thank the program committee for delivering high-quality reviews on a very tight schedule: you guys are the best.

Sharon O'Brien, Michel Simard and Lucia Specia

Oral Presentations and Posters

The CRITT TPR-DB 1.0: A Database For Empirical Human Translation Process Research

Michael CARL

Post-editing Time as a Measure of Cognitive Effort

Maarit KOPONEN, Wilker AZIZ, Luciana RAMOS and Lucia SPECIA

Average Pause Ratio as an Indicator of Cognitive Effort in Post-editing: A Case Study

Isabel LACRUZ, Gregory M. SHREVE and Erik ANGELONE

Reliably Assessing the Quality of Post-edited Translation Based on Formalized Structured Translation Specifications

Alan K MELBY, Jason HOUSLEY, Paul J FIELDS and Emily TUIOTI

Learning to Automatically Post-edit Dropped Words in MT

Jacob MUNDT, Kristen PARTON and Kathleen McKEOWN

SmartMATE: An Online End-To-End MT Post-editing Framework

Sergio PENKALE and Andy WAY

To post-edit or not to post-edit? Estimating the benefits of MT post-editing for a European organization

Alexandros POULIS and David KOLOVRATNIK

How Good is Crowd Post-editing? Its Potential and Limitations

Midori TATSUMI, Takako AIKAWA, Kentaro YAMAMOTO and Hitoshi ISAHARA

Demos

Organizers

Sharon O'Brien – CNGL / DCU
Michel Simard – National Research Council Canada
Lucia Specia – University of Sheffield

Program Committee

Nora Aranberri – TAUS
Diego Bartolome – tauyou
Louise Brunette – Université du Québec en Outaouais
Michael Carl – Copenhagen Business School
Francisco Casacuberta – Universitat Politècnica de València
Mike Dillinger – Translation Optimization Partners
Stephen Doherty – Dublin City University
Andreas Eisele – European Commission
Jakob Elming – Copenhagen Business School
Atefeh Farzindar – NLP Technologies
Marcello Federico – FBK-IRST
Mikel L. Forcada – Universitat dAlacant
Ana Guerberof – Logoscript
Nizar Habash – Columbia University
Daniel Hardt – Copenhagen Business School
Kristian Tangsgaard Hvelplund – Copenhagen Business School
Pierre Isabelle – National Research Council Canada
Maxim Khalilov – TAUS
Philipp Koehn – University of Edinburgh
Roland Kuhn – National Research Council Canada
Philippe Langlais – RALI / Université de Montréal
Alon Lavie – Carnegie Mellon University
Elliott Macklovitch – Translation Bureau Canada
Daniel Marcu – SDL-LW / USC / ISI
John Moran – Transpiral Translation Services
Kristen Parton – Columbia University
Maja Popovic – DFKI
Alexandros Poulis – European Parliament/Intrasoft
Johann Roturier – Symantec

Jean Senellart – SYSTRAN
Myriam Siftar – MTM LinguaSoft
Roberto Silva – Celer Soluciones
Radu Soricut – SDL International
Midori Tatsumi – Dublin City University
Jörg Tiedemann – Uppsala University
Andy Way – Applied Language Solutions
Chris Wendt – Microsoft Research

The CRITT TPR-DB 1.0:
A Database for Empirical Human Translation Process Research

Michael Carl
Institute for International Business Communication
Copenhagen Business School,
2000 Frederiksberg, Denmark
mc.ibc@cbs.dk

Abstract

This paper introduces a publicly available database of recorded translation sessions for Translation Process Research (TPR). User activity data (UAD) of translators behavior was collected over the past 5 years in several translation studies with Translog [1], a data acquisition software which logs keystrokes and gaze data during text reception and production. The database compiles this data into a consistent format which can be processed by various visualization and analysis tools.

1 Introduction

Human translation process research (TPR) is a branch of descriptive translation studies (Holms, 1972) which analyzes the translation behavior of translators, such as types of units that translators focus on, conscious and unconscious translation processes, differences in expert and novice behavior, memory and search strategies to solve translation problems, etc. It seeks to identify the temporal (and/or contextual) structure of those activities and describes inter- and intra-personal variation. Various models have been developed that seek to explain translators' behavior in terms of controlled and uncontrolled workspaces (Göpferich, 2008), and monitor models (e.g. Tirkkonen-Condit, 2005) with trigger micro- and macro-translation strategies. However, due to the lack of appropriate data and tools, only few attempts have been made to ground and quantify translation process models in empirical user activity data (UAD).

In order to close this gap, this paper introduces a database of translation process data which was collected over the past 5 years with Translog[1]. More than 450 translation sessions were recorded in 10 translation studies and converted into a common format (Carl and Jacobsen, 2009). The database is now publicly available, together with a toolkit for analysis and visualization: as described in Carl and Jacobsen, (2009), the UAD consists of product and process components which are processed in different components in the CRITT TPR-DB[2]. A) We used the NLTK (Bird, 2009)[3] for automatically POS tagging and lemmatization. B) In addition, the product data can be converted into treex format and visualized/annotated in TrEd[4]. C) The CRITT TPR-DB provides several tools to manually check and amend the automatic annotations. D) The product and process data is integrated by mapping keystrokes and fixations on the produced TT tokens (Carl, 2012) and via the alignment on the corresponding ST equivalents. This allows us to extract various different types of product and process units from the UAD and to mutually correlate the product and the process data. Translation sessions can thus be visualized in

[1] The translog website is www.translog.dk. The most recent version of Translog-II can be obtained for free for academic purposes from the author.

[2] CRITT (www.cbs.dk/en/CRITT) is the "Center for Research and Innovation in Translation and Translation Technology" at Copenhagen Business School. We refer to the UAD database as CRITT TPR-DB.

[3] NLTK is a Python platform to work with human language data: http://nltk.org/

[4] TrEd is a programmable graphical editor and viewer for tree-like structures: http://ufal.mff.cuni.cz/tred/

the form of translation progression graphs (Carl and Jacobsen, 2009) or statistically analyzed e.g. with R[5].

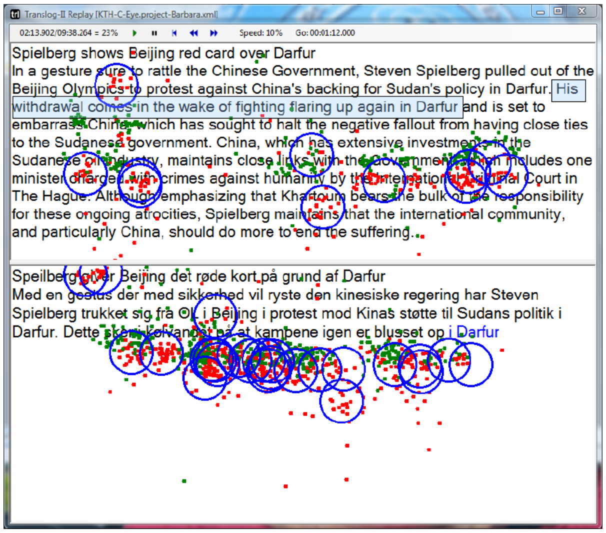
In this paper we give a short introduction to translation process research and the data that we obtain from Translog. We describe the structure of the CRITT TPR-DB and the origin/intention of the various studies it contains. We will then describe how the raw logging data is compiled into a database structure which allows for more detailed analysis and evaluation of the translation processes. While much of this compilation is fully automatized, the database design also contains a number of tools to manually adjust the annotations. Finally we give an overview of the Metadata that is stored with the CRITT TPR-DB.

2 Empirical TPR with Translog

While in the beginnings of TPR, user activity data (UAD) could only be elicited via traditional methods of introspection such as questionnaires, think-aloud experiments (TA) or retrospection (Krings, 1986; Lörscher, 1992; Tirkkonen-Condit & Jääskeläinen, 2000), computer-based analysis techniques have been applied in empirical translation studies for about 15 years.

Around the 1990s, most texts and most translations were typed on computer keyboards, and software was developed to log the writing process (all keystrokes, pauses and changes), for example ScriptLog (Holmqvist et al, 2002), Proxy (Pacte group), Translog (Jakobsen and Schou, 1999 and Inputlog (Leijten/Van Maes, 2006)). This can be regarded as the beginning of digital translation process research (DTPR). With these tools a complete log can be created of all the keystrokes made in producing a text, including typos, pauses, deletions, changes, mouse clicks, cursor movements. Several larger translation process projects were carried out with keystroke logging combined with retrospection and post-process dialogues.

Since 2006 CRITT [6] has developed a data acquisition software, Translog (Jakobsen and Schou, 1999, Carl 2012) with which translators' keystroke and gaze activities can be recorded[7]. This tool is now the most widely used tool of its kind (Jakobsen, 2006).



Spielberg shows Beijing red card over Darfur
In a gesture sure to rattle the Chinese Government, Steven Spielberg pulled out of the Beijing Olympics to protest against China's backing for Sudan's policy in Darfur. His withdrawal comes in the wake of fighting flaring up again in Darfur and is set to embarrass China which has sought to halt the negative fallout from having close ties to the Sudanese government. China, which has extensive investments in the Sudanese oil industry, maintains close links with the Government, which includes one minister charged with crimes against humanity by the International Criminal Court in The Hague. Although emphasizing that Khartoum bears the bulk of the responsibility for these ongoing atrocities, Spielberg maintains that the international community, and particularly China, should do more to end the suffering.

Speilberg giver Beijing det røde kort på grund af Darfur
Med en gestus der med sikkerhed vil ryste den kinesiske regering har Steven Spielberg trukket sig fra OL i Beijing i protest mod Kinas støtte til Sudans politik i Darfur. Dette sker i kølvandet på at kampene igen er blusset op i Darfur

Figure 1: Screenshot of Translog-II replay: fixations in blue circles

As shown in figure 1, Translog separates the screen into two windows: the source text is shown in the upper window while subjects type a translation into the lower window. Figure 1 also shows the accumulations of gaze fixations (in blue) during the time span in which a translator reads the beginning of the source language sentence “China which has extensive investments in the Sudanese oil industry, maintains close” and begins producing (i.e. typing in) its translation.

Translog-II can be used to record reading and writing activities, as well as sessions of post-editing and revision. For post-editing (e.g. of MT output), the translation session can be prepared in such a way that the translation to be revised appears in the lower window of the screen while the upper window contains the original source text. Writing studies would be initiated by preparing Translog-II to show only the lower window, and reading experiments would plot only the upper window. In a similar way, a revision (or editing) scenario of a text without a source can be produced by plotting the lower (write enabled) window with

[5] R is a free software environment for statistical computing and graphics. It can be downloaded from http://www.r-project.org/

[6] CRITT aims at building up new knowledge of translation and communication processes and provide a basis for technological innovation in this field.

[7] Translog-II has interfaces to Tobii eye-tracker; a connection to eye-link 1000 is currently being implemented.

a pre-defined text. Note that the screen can also be divided in a vertical manner.

3 Translation Process Database

CRITT has collected over the past 5 years a substantial amount of translation process data from numerous translation sessions. The analysis of this data has given rise to more grounded translation models and an extended understanding of the underlying human translation processes (Mees and Göpferich, 2009, Göpferich, Jakobsen, Mees, 2009; Göpferich, Alves, Mees, 2010).

As the collected UAD was recorded with various Translog versions producing different logging formats, the data has been converted into one consistent data format (Carl and Jakobsen, 2009) and annotated with Metadata (Jensen and Carl, 2012). In addition, more than 230 translation sessions were recorded in the past year to complement the legacy TPR UAD with more target languages and with post-editing sessions. In its current version, the CRITT TPR-DB consists of 10 translation studies which amount to a total of 456 (translation) sessions, distributed as follows:

T: 257 Translation (from scratch)
P: 129 Post-editing
E: 40 Editing
C: 30 Text Copying

Table 1: Distribution of recordings per Study and ST in the CRITT TPR-DB V1.0: lines represent different Studies, rows different source texts

Study | Text	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	Total
ACS08				14	16	15	15													60
BD08								10												10
BML12	9	11	10					10										10	10	60
JLG10												2	3	2	3	5	5			20
KTHJ08	24	24	23																	71
LWB09									12	14	14									40
MS12	3	9	7					10										8	7	44
NJ12	15	19	14					17										18	17	100
SG12	6	5	5					6										5	5	32
TPR11	10		9																	19
Total translations	67	69	67	14	16	15	15	53	12	14	14	2	3	2	3	5	5	41	39	456

In each session, a translator had to translate (T), post-edit (P), Edit (E) or copy (C) a source text. In the case of post-editing, MT output was shown in the target window, and in the case of editing the MT output was shown without the source text (monolingual editing of MT output). A total of 19 different source texts were used in these studies, so that there are on average 24 translations per text. Table 1 shows the distribution of translations for each source text. While some texts (Text1, Text2, Text3 and Text8) have been translated more than 50 times into various languages and have been re-used in several translation studies, other texts are translated only few times. Text12, Text13, Text14 and Text15 are only used in one study and have been translated only by 2 and 3 translators respectively.

Each source text is between 100 and up to 236 words in length and designed in a way such that it fits on one Translog screen (to avoid scrolling). 13 of the 19 source texts are English, and two translation studies, JLG10 and LWB08, use respectively Portuguese and Danish source texts to be translated into English. Some of the source texts only differ in few words, as they seem to be slightly modified in some experiments.

With respect to the target languages, the CRITT TPR-DB is more varied than with the source languages, with a total of 7 different target languages. The table 2 shows the distribution of translation, post-editing, editing and copying experiments together with the respective source and target languages. Note that the source language is also given in the editing experiments (even though the text was not visible for the editor) and that copying experiments have identical source and target languages.

Table 2: Distribution of recordings with respect to source and target language and type of session.

Source	Target	T	P	E	C	Total
en	da	111				111
en	hi	39	61			100
en	es	20	20	20		60
en	zh	15	19	10		44
en	de	12	19	10		41
da	en	40				40
en	en				30	30
en	pt	10	10			20
pt	en	10				10

With the exception of study JLG10 (20 translation sessions), all of the studies contain keystroke and gaze data. Gaze data was collected with Tobii eyetracker 1750 (BD08, ACS08, KTHJ09 and LWB09), Tobii T120 (TPR11, BML12, MS12, NJ12) and Tobii TX300 for SG12. The 10 studies were conducted for different reasons and with different research goals. While the collected data has been evaluated in numerous publications, the primary purpose of the studies were as follows:

ACS08: 30 translations (en->da) and 30 text copying sessions (en->en). The aim of this study was to explore the way in which translators process the meaning of non-literal expressions (Sjørup, 2011)

BD08: 10 translations (en->da), collected in the context of the Eye-to-IT project, to investigate production pauses (Dragsted, 2010)[8].

KTHJ08: 72 translations (en->da) to investigate translators' allocation of cognitive resources (Jensen, 2011).

LWB09: 40 translations (da->en) to investigate the impact of syntactic processing in translation from L1 to L2 (Sjørup et al. 2009)

JLG10: 10 translations en->pt and 10 translations pt->en to investigate the impact of direct (L2-L1) and indirect (L1-L2) translations. (Gonçalves and Alves, 2012)

TPR11: 10 post-editing sessions en->pt and 9 post-editing sessions en->de collected in the context or the TPR summer school 2011.

The following four studies were conducted in the context of the CASMACAT[9] project, with the aim to compare translation, post-editing and editing activities. A set of 6 English texts was translated and post-edited into Spanish, Chinese, Hindi and German.

BML12: 20 translation, 20 post-editing and 20 editing sessions, all en->es (Mesa-Lao, 2012)

MS12: 15 translation, 19 post-editing and 10 editing sessions, all en->zh (Schmalz, 2012)

NJ12: 39 translation and 61 post-editing sessions, all en->hi (Jaiswal et al. 2012)

SG12: 12 translation, 10 post-editing and 10 editing sessions, all en->de (Hansen and Gutermuth, forthcoming)

4 Database Compilation

The collected TPR UAD is processed and annotated to allow for more detailed analysis and evaluation of the translation processes. For each of the logging files a compilation process produces the following four types of resources (in several different different files) which, in addition to the metadata, constitute the CRITT TPR-DB 1.0:

1. Logged UAD (output of Translog)
2. Aligned and annotated product data
3. Treex representations of the product data
4. Unit tables for (quantitative) analysis and visiualization of translation progression graphs

[8] http://cogs.nbu.bg/eye-to-it/

[9] http://www.casmacat.eu/

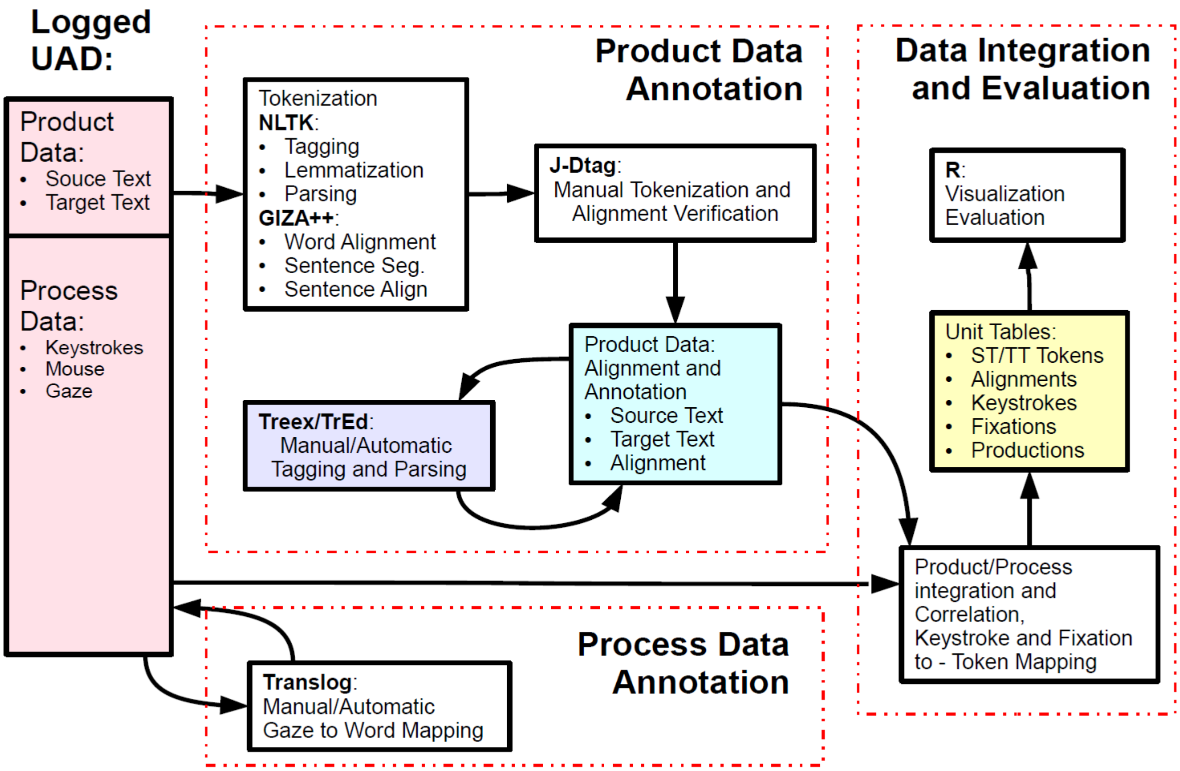


Figure 2: Diagram for the compilation of CRITT TPR-DB V1.0: from the logged UAD is semi-automatically generated 1. aligned and annotated product data, 2. treex representations and 3.unit tables.

Note that the CRITT TPR-DB follows a consistent naming strategy for the folders and files. To annonymise the recordings, filenames consist of a naming strategy which enumerated the participant, the task (translation, post-editing, etc.) and the text. Thus, a recording with the file root *P02_T1* e.g. in BD08 would refer to the recording of participant no. 2 (*P02*) for a translation task of text 1 (*T1*) in that particular study. This file root is kept consistent for all derived and annotated information for this recording. The concatenation of the study name and the file root – e.g. *BD08P01T1* - thus gives a unique identifier for a recording.

Figure 2 plots the processing steps in which the CRITT TPR-DB 1.0 is generated while Figure 3 shows the structure of the database. Besides the studies folders, the database also contains a Treex, a MetaData, and a bin folder.

Following the description in Carl and Jakobsen (2009), a distinction is made between product data and process data. Figure 2 shows that both types of data are, to a certain extent, processed independently and then integrated for the production of unit tables. This information is stored under the Study folder in separate subfolders. The product data (i.e. the final source and target texts) are extracted from the Translog-II logging protocol and linguistically processed in the following steps:

1. Tokenization
2. Sentence segmentation
3. Sentence alignment
4. Word alignment
5. POS tagging and Lemmatization
6. Dependency annotation

Tokenization and sentence segmentation is processed based on our own tools[10], while sentence and word alignment was pre-processed with Giza++ and manually checked and corrected for all of the 456 translation sessions. POS tagging and lemmatization alignment was achieved with the tree tagger for German, English, Danish. We plan

[10] Chinese Tokenization was manually corrected based on a tool provided by Derek Fai Wong, University of Macao.

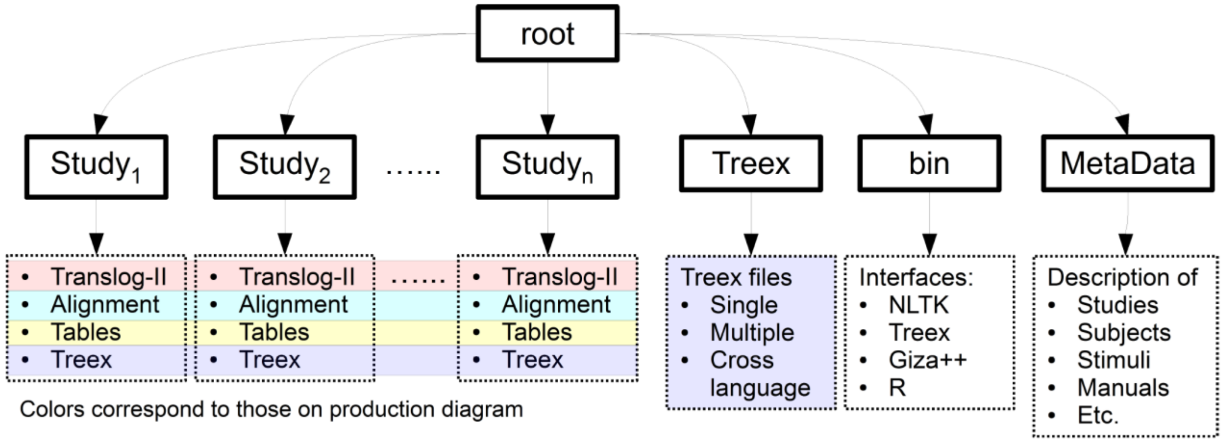


Figure 3: Representation of the CRITT TPR-DB V1.0: the initial Translog-II logging data is enriched with alignments and annotations, as well as with MetaData. Further studies and recordings can be added and processed by a set of programs and scripts in the bin folder.

to manually annotate dependency relations for all source files, as well as for all the sessions in the target files of BD08 study, using the DTAG annotation schema[11]. The TPR-DB product data is also represented in the Treex format to be visualized in TrEd and to manually correct the linguistic annotation. The Treex folder contains two types of treex representations:

Table 3: example of alignment units (AU) table showing source and target unit with, the typed string, length of the typed sequence (insertions, deletions), as well as starting time and pre-unit production pause.

AUtarget	AUsource	Len	Ins	Del	Time1	Pause1	Typed
Selvom	Although	7	7	0	1267	12395	Selvom_
udviklingslande_forståeligt	developing_countries	34	31	3	7414	3029	udviklingl[l]slande_forståelig…
er_nok	are_understandably	7	7	0	688	142	nok_er_
tilbageholdende_med	reluctant	32	26	6	17525	841	tilbageholdende_[_edned]dend…
at	to	65	34	31	61505	89	at_gå_på_kompromis_med[de…
ødelægge	compromise	9	9	0	2156	5767	ødelægge_
deres	their	6	6	0	847	120	deres_
chancer_at	chances	11	11	0	1026	237	chancer_at_
for_opnå	of	9	9	0	343	128	for_opnå_

- For each recording a separate treex file is generated, containing only the source text and one translation
- For every source text one treex file is generated, containing all translations for this text.

There are thus 456 treex files of the former and 19 treex files of the latter type.

The annotated product data is integrated with the process data by mapping keystrokes and fixations - which occur during the text production - on the source and target language tokens that are being typed or gazed at. The underlying algorithms are described in (Carl and Jakobsen, 2009) and an updated version is available in (Carl, 2012). The integration of the product and process data allows us to generate various unit tables which can then be analyzed and visualized, for instance with R. Currently, the following seven unit tables are produced, each line describes:

Source tokens: enumeration of ST token

Target tokens: enumeration of TT token together with ST correspondence, number, time and value of production keystrokes (number of insertions and deletions).

[11] http://code.google.com/p/copenhagen-dependency-treebank/

Keystrokes: text modification (insertions or deletions), together with time of stroke, and the word in the final text to which the keystroke contributes.

Fixations: starting time, end time and duration of fixation, as well as character offset and word id of fixated symbol in the source or target window.

Production units: starting time, end time and duration of coherent sequence of typing (cf. Carl and Kay, 2011), percentage of parallel reading activity during unit production, duration of production pause before typing onset, an well as number of insertion, deletions.

Fixation units: starting time, end time and duration of coherent sequence of reading activities as defined in (Carl and Kay, 2011), as well as ids of fixated words.

Alignment units: source and target correspondences of AU, number of production keystrokes (insertions and deletions) duration of production and revision time, amount of parallel reading activity during AU production.

Each of the units is characterized by a number of features with a consistent naming strategy, so as to easily map contents of different tables. Table 3 in an example of alignment units table: each line describes an AU with a number of features. The data can be statistically evaluated (e.g. with R, for which various scripts exist) for quantitative analysis of translation processes. Given the richness of the CRITT TPR-DB and the structured representation of the data, a large number of additional features may be generated with little effort. Future evaluation of the data will generate needs for additional features which can be easily integrated in the existing framework.

5 Manual Correction

Manual correction and verification of the automated annotation processes are important at all levels of representation. The CRITT TPR-DB compilation process anticipates several steps to manually interfere and checking mechanism are put in place to ensure that the data remains consistent. Currently there are three programs

Jdtag: is a java implementation of a simplified version for bilingual alignment which is compatible with the dtag tool (Kromann, 2003). It allows to visualize word alignments and to modify alignment information in a command line[12], as shown in figure 4.

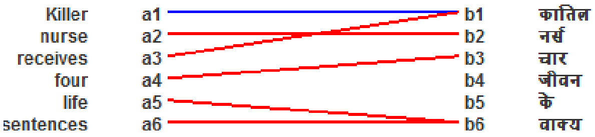


Figure 4: example of alignment visualization in Jdtag

Treex and TrEd: are free software distributed under GPL. TrEd is a fully customizable and programmable graphical editor and viewer for tree-like structures which runs on windows and Unix platforms. The conversion makes use of the Treex[13] programming interface. Figure 5 shows an example of the GUI.

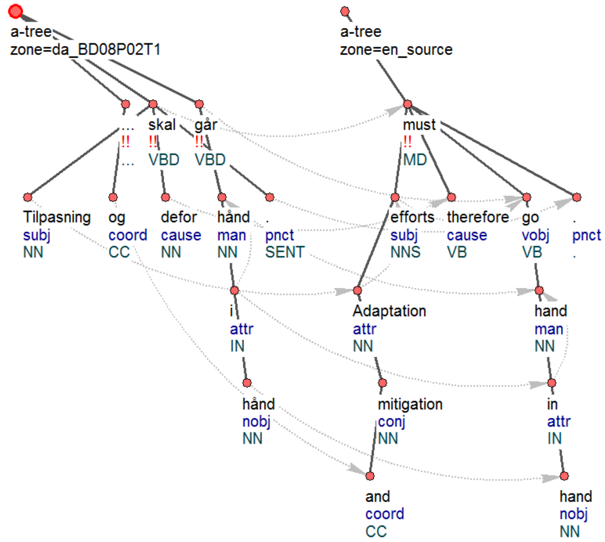


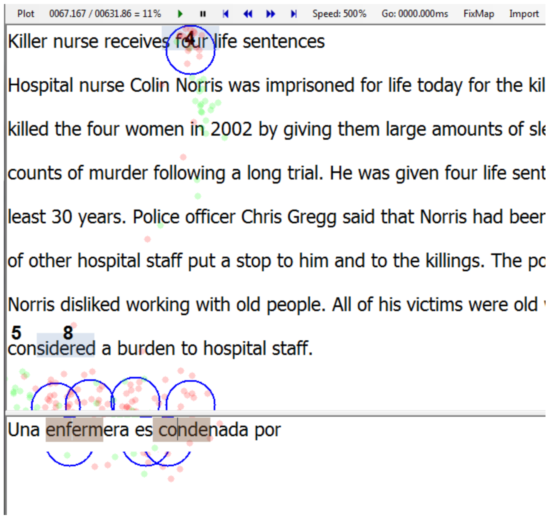
Figure 5: Example of dependency tree alignment and annotation in TrEd

Translog-II: While there are a number of tools and approaches to manually inspect, annotate and amend the product data (such as dtag, Jdtag and TrEd) there are only very few tools for annotating process data, such as the LITTERAE search tool (Alves & Vale 2011). Manual correction of process data includes amendment of logging errors, and the adjustment of gaze-to-word mapping. Due to free head movement and other sources of noise, calibration of gaze data gets often imprecise, so that the captured fixations often cannot be simply mapped to the closest underlying symbols. Despite a font size of 17pt, which was usually chosen in the translation studies, we frequently observe fixation drift to the next line. As shown in Figure 6, we implemented an additional replay mode (FixMap) in the Translog-II program which allows to manually re-

[12] Jdtag was implemented by Ragnar Bonk. It is free software that can be downloaded upon request.

[13] http://search.cpan.org/~tkr/Treex-Doc-0.08324/lib/Treex/Tutorial/Install.pod

assign fixation mappings during the replay of translation sessions, and to store the amended file under a different name.



Killer nurse receives four life sentences

Hospital nurse Colin Norris was imprisoned for life today for the kil

killed the four women in 2002 by giving them large amounts of sle

counts of murder following a long trial. He was given four life sent

least 30 years. Police officer Chris Gregg said that Norris had beer

of other hospital staff put a stop to him and to the killings. The pc

Norris disliked working with old people. All of his victims were old

considered a burden to hospital staff.

Una enfermera es condenada por

Figure 6: manual fixation correction in Translog-II: erroneous gaze-to-word mapping caused by gaze drift of can be manually.

6 Meta Data

The MetaData folder (see Figure 1) contains very detailed meta data information, as proposed in (Jensen and Carl, 2012). It consists of four csv files:

1. **Study MetaData**: enumerates the studies in the database, describes the purpose of the study, including a bibliography. It contains five categories of information:

- ExperimentID is a unique identifier which is represented as a derived element in Stimulus metadata and Recordings metadata.
- Abstract contains an abstract of the main study for which the process data have been collected.
- Keywords lists the keywords of the experiment.
- MainLiterature contains a reference to the main study for which data have been collected.
- SecondaryLiterature contains references to other studies than the main study that have analysed data from the experiment.

2. **Stimulus MetaData**: describes the static properties of the source texts used in the study, their length, domain, etc. It contains the following categories of information:

- StimulusID is a unique identifier which is represented as a derived element in Recordings metadata.
- SourceLanguage states the language of the source text.
- LengthWords states the number of words of the source text.
- LengthCharacters states the number of characters of the source text.
- Text contains the source text in its entirety.

3. **Recordings MetaData**: provides background for the recordings, such as which texts were used, which hard and software configuration, source and target languages, and date of the recording etc.

- EyeTrackerType specifies the eye tracking equipment that was used to collect the eye-tracking data.
- RecordingSoftware specifies the eye tracking recording software that was used to collect the eye-tracking data.
- EyeTrackerSoftwareVersion specifies the software version of the eye-tracking recording software.
- Keylogger specifies the keylogging software that was used to collect the typing data.
- KeyloggingSoftwareVersion specifies the software version of the keylogging software.
- ExperimentalLocation specifies where the recording was carried out.
- TargetLanguage specifies the language into which the source text was translated, copied, post-edited, etc

4. **Participants MetaData**: contains information about the participants from whom process data have been collected. It contains the following information:

- ExperimentID is a derived identifier from Study metadata which links the participant explicitly to an experiment.
- ExperimentParticipantID is a unique identifier which is represented as a derived element in Recordings metadata.
- Sex of the participant.
- YearOfBirth of the participant.
- Programme that the participant was enrolled into.
- Student at the time of recording (yes/no).
- DegreeStartedYear specifies the year in which the participant was enrolled into a university programme.
- DegreeFinishedYear specifies the last year of the participant's university programme enrolment.
- YearsTraining specifies the number of years the participant received translation specific instruction.

- CertifiedTranslator specifies whether or not the participant has received formal authorisation to work as a translator and/or interpreter.
- ExperienceYears specifies the number of years the participant has worked as a professional translator.
- L1 of the participant.
- L2 of the participant.
- L3 of the participant.
- OpticalAids specifies whether or not the participant uses optical aids such as glasses or contact lenses.
- LeftEye specifies the dioptre for the left eye.
- RightEye specifies the dioptre for the right eye.
- EyeColour of the participant.

.

Note that not all information is provided for all studies/participants/recordings. In fact it is difficult to gather all the data for experiments which have been conducted 5 years ago. While the naming convention in the Metadata is consistent with the study and recording name in as described in section 4, there is, as of now, no appropriate query tool available.

7 Conclusion

The paper describes the first public release of the CRITT TPR-DB. More than 450 translation sessions were recorded (more than 400 with gaze data) linguistically annotated and stored in a consistent data format. The database contains translations mainly from English into very different languages, such as Spanish, Hindi, Chinese and German, produced by novice and experienced translators. It contains from scratch translations, mono- and bilingual post-edited MT output (google and AnglaBharati (Sinha, 2005)) as well as text copying, with very detailed key logging and gaze data information. Some of the data also has detailed metadata information about the Stimulus, Recording and Participant. It is thus possible to compare translation behavior of the same participant across different studies and tasks (translation, post-editing, etc.) as well as compare translation strategies of different translators when translating the same text into different languages.

In future releases of the database we will add more experiments, complete the annotation (e.g. by adding more dependency annotations), but also add more tools to query the database and extract more features for the unit tables. Particular focus will also be given to the gaze data and gaze-to-word mapping strategies, as this seems to be the most noisy and least understood part in the database. Given the increased interest in post-editing, we hope that the CRITT TPR-DB will attract researchers to analyze and compare translation and post-editing processes to better understand and model these different activities, and to finally develop tools that better support translators in their work.

Acknowledgments

This collection would not have been possible without the year-long work of the experimenters as well as the many study participants, translators, translation students and bilingual editors and annotators. In particular thanks to Zdenek Zabokrtsky and Jan Stephanek for helping with TrEd and Treex, to Abhijit Mishra who implemented the NLTK interface during a summer internship and Kristian TH Jensen for his ideas on the MeteData.

References

Alves, F. & Vale, D. C. 2011. *On drafting and revision in translation: a corpus linguistics oriented analysis of translation process data,* Translation: Computation, Corpora, Cognition, Vol 1, No 1 http://www.t-c3.org/index.php/t-c3/article/view/3

Steven Bird, Ewan Klein, and Edward Loper . Natural Language Processing with Python --- Analyzing Text with the Natural Language Toolkit

Carl, Michael (2012). Translog-II: a Program for Recording User Activity Data for Empirical Reading and Writing Research, In Proceedings of the Eight International Conference on Language Resources and Evaluation, European Language Resources Association (ELRA)

Carl, M. and Jakobsen A.L. (2009). Towards statistical modelling of translators' activity data. International Journal of Speech Technology, 12(4). http://www.springerlink.com/content/3745875x22883306/.

Carl, M., and Kay, M. (2011). Gazing and Typing Activities during Translation: A comparative Study of Translation Units of Professional and Student Translators, META 56:4

Dragsted, B (2010). Co-ordination of reading and writing processes in translation. Contribution to Translation and Cognition, Shreve, G. & Angelone, E. (eds.). John Benjamins in 2010.

Gonçalves, José Luiz V. R. and Alves, Fabio, Investigating the conceptual-procedural distinction in the translation process: a relevance-theoretic analysis of micro and macro translation units, in press

Göpferich, S., Jakobsen, A. L., and Mees, I. M., editors (2009). Behind the Mind: Methods, Models and Results in Translation Process Research. Copenhagen: Samfundslitteratur, Copenhagen

Göpferich, Susanne (2008): Translationsprozessforschung: Stand - Methoden - Perspektiven. Translationswissenschaft 4. Tübingen: Narr

Göpferich, Susanne/Alves, Fabio/Mees, Inger M., Hrsg. (2010): New Approaches in Translation Process Research. Copenhagen Studies in Language 39. Kopenhagen: Samfundslitteratur.

Hansen and Gutermuth (2013), forthcoming

Holmes, James S. (1972/1988). The Name and Nature of Translation Studies. In Holmes, Translated! Papers on Literary Translation and Translation Studies, Amsterdam: Rodopi, pp. 67–80.

Jaiswal, Sinha and Shukla (2012), Analysing User Activity Data of English-Hindi Translation, forthcoming

Jakobsen, A.L. und Schou, L. (1999). “Translog documentation.” Copenhagen Studies in Language 24: 149–184.

Jakobsen, A.L. (2006). “Research methods in translation – Translog.” In Sullivan, K. P. H. and Lindgren, E., (eds) Computer keystroke logging and writing: Methods and applications, volume 18. Oxford: Elsevier, 95–105

Jensen, HKT and Carl, Michael. User Activity Metadata for Reading, Writing and Translation Research. In: Proceedings of The Eighth International Conference on Language Resources and Evaluation. LREC 2012: Workshop: Describing LRs with Metadata: Towards Flexibility and Interoperability in the Documentation of LR. ed. / Victoria Arranz; Daan Broeder; Bertrand Gaiffe; Maria Gavrilidou ; Monica Monachini ; Thorsten Trippel. Paris : ELRA, 2012. p. 55-59.

Jensen, Hvelplund, K. T. (2011), Allocation of cognitive resources in translation: an eye-tracking and key-logging study. Ph.d.-afhandling. Copenhagen Business School. Copenhagen.

Krings, H.P. (1986). Was in den Köpfen von Übersetzern vorgeht. Tübingen: Narr.

Kromann, M. T. (2003). The Danish Dependency Treebank and the DTAG treebank tool. In Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003), 14-15, November, Växjö. 217–220.

Van Maes, L. and Leijten, M. (2006). “Logging writing processes with Inputlog.” In Van Maes, L., Leijten, M. and Neuwirth, D. (eds). Writing and Digital Media. Oxford: Elsevier, 158-165.

Lörscher, W. (1992). “Investigating the Translation Process.” Meta, XXXVII (3):426–439.

Mesa-Lao, Barto (2013), Analysing User Activity Data of English-Spanish Translation, International Workshop on Expertise in Translation and Post-editing Research and Application, Copenhagen

Mees, I. M., Alves, F. and Göpferich, S., (eds) (2009). Methodology, Technology and Innovation in Translation Process Research, volume 38. Copenhagen: Samfundslitteratur.

Sinha R.M.K., (2005), Integrating CAT and MT in AnglaBharti-II Architecture , EAMT 2005, Budapest, Hungary.

Sjørup, Annette C. Cognitive Effort in Metaphor Translation : An Eye-Tracking Study.. I: Cognitive Explorations of Translation. red. / Sharon O'Brien. London : Continuum International Publishing Group Ltd, 2011. s. 197-214 (Continuum Studies in Translation).

Sjørup, A.C., Jensen, K.T.H., & Balling, L.W. (2009). Syntactic processing in translation from L1 to L2. Eye-to-IT conference on translation processes 28-29 April 2009, Copenhagen, Denmark.

Schmalz (2012). English-Chinese Manual and Post-Ed Translation Process: a Comparative Pilot Study, International Workshop on Expertise in Translation and Post-editing Research and Application, Copenhagen

Tirkkonen-Condit,Sonja (2005) The Monitor Model Revisited: Evidence from Process Research, META, Volume 50, numéro 2, avril 2005, p. 405-414

Tirkkonen-Condit, S. & Jääskeläinen, R. (eds) (2000). Tapping and Mapping the Process of Translation and Interpreting. Amsterdam & Philadelphia: John Benjamins.

Post-editing time as a measure of cognitive effort

Maarit Koponen
Dept of Modern Languages
University of Helsinki
maarit.koponen@helsinki.fi

Wilker Aziz
RIILP
University of Wolverhampton
w.aziz@wlv.ac.uk

Luciana Ramos
Scientific Translator and Interpreter
and Professional Trainer
translationandtraining@gmail.com

Lucia Specia
Dept of Computer Science
University of Sheffield
l.specia@sheffield.ac.uk

Abstract

Post-editing machine translations has been attracting increasing attention both as a common practice within the translation industry and as a way to evaluate Machine Translation (MT) quality via edit distance metrics between the MT and its post-edited version. Commonly used metrics such as HTER are limited in that they cannot fully capture the effort required for post-editing. Particularly, the cognitive effort required may vary for different types of errors and may also depend on the context. We suggest post-editing time as a way to assess some of the cognitive effort involved in post-editing. This paper presents two experiments investigating the connection between post-editing time and cognitive effort. First, we examine whether sentences with long and short post-editing times involve edits of different levels of difficulty. Second, we study the variability in post-editing time and other statistics among editors.

1 Introduction

As Machine Translation (MT) becomes widely available for a large number of language pairs and the demand for faster and cheaper translations increases, its adoption is becoming more popular in the translation industry. However, it is well known that except in very narrow domains with dedicated MT systems, automatic translations are far from perfect. A common practice is thus to have human translators performing post-editing of such translations. Post-editing has also been attracting increasing attention from researchers in MT as a way of evaluating the quality of machine translations, particularly for the purpose of comparing various MT systems. Effective ways of measuring post-editing effort – and thus MT quality – in both scenarios is a very relevant but open problem.

Standard MT evaluation metrics have proved to correlate significantly better with human assessments of quality when computed having a post-edited version of the automatic translation as reference as opposed to translations created independently from automatic translations. One of these metrics is HTER – the “Human-targeted Translation Edit Rate” – (Snover et al., 2006), which was used as the official metric in the DARPA GALE program (Olive et al., 2011).

HTER is an edit distance metric that computes the minimum number of edits between the system output and its (often minimally) post-edited version. It is a simple metric which has nevertheless shown to be very effective. However, this and other metrics that estimate the similarity or distance between a system translation and its post-edited version have a crucial limitation: they cannot fully capture the effort resulting from post-editing such a translation. Certain operations can be more difficult than others, based not only on the type of edit (deletion, insertion, substitution), but also on the words being edited. Edits due to incorrect morphological variants or function words are generally treated the same way as more complex edits such as fixing an untranslated content word. While variants of such metric assigning weights for specific edits or classes of words can be implemented (Snover et al., 2010; Blain et al., 2011), defining classes of complex words to post-edit requires a lexicalised, linguistically- motivated and thus language-dependent approach. In addition, the complexity of a correction cannot always be characterized based only on a local edit, as it may depend on the neighbourhood of that edit.

Recently, Koponen (2012) conducted an error analysis on post-edited translations with HTER and 1-5 scores assigned by humans for post-editing effort. A number of cases were found where post-edited translations with low HTER (few edits) were assigned low quality scores (high post-editing effort), and vice-versa. This seems to indicate that certain edits require more cognitive effort than others, which is not captured by HTER.

Post-editing effort consists of different aspects: temporal, technical and cognitive (Krings, 2001). However, these aspects are highly interconnected. The temporal effort (time spent on post-editing) is the most easily measurable. Post-editing time reflects not only the technical effort needed to perform the editing, but also the cognitive effort required to detect errors and plan the necessary corrections.

We believe that measuring *post-editing time* is the most cost effective and straightforward way of quantifying at least some of the cognitive effort involved in post-editing. In order to verify this hypothesis, in this paper we study measurements of post-editing time on a number of English-Spanish translations produced by eight MT systems and revised by eight translators. We follow a similar methodology as Koponen (2012), but focus on discrepancies between post-editing *time* and HTER. The main purpose of this experiment is to identify different groups of errors in MT and correlate them to different levels of difficulty that may be involved in fixing them, where difficulty is defined in terms of post-editing time. We are particularly interested in sentences that take a long time to edit but involve relatively few edit operations (low HTER) and are not excessively long. In addition, we use time and other detailed post-editing effort indicators, such as number of keystrokes, to analyse the variance between different translators post-editing the same translations.

The remainder of this paper is organized as follows. Section 2 presents previous attempts to measure post-editing effort. Section 3 describes the dataset and method used for our analysis and Section 4 shows the results of this analysis.

2 Related work

The most commonly used approach to quantifying post-editing effort (and in general translation quality) has been the use of semi-automatic MT evaluation metrics such as HTER that measure the similarity or distance between the MT system output and its human post-edited version. However, while such metrics provide a good indication of the technical effort involved in post-editing, they do not capture its cognitive effort. Koponen (2012) shows that translator's perception of post-editing effort, as indicated by scores in 1-5, does not always correlate well with edit distance metrics such as HTER. In other words, sentences scored as requiring significant post-editing sometimes involve very few edits, and vice-versa. What this suggests is that technical and cognitive effort are not always equal: certain types of errors require considerable cognitive effort although the number of technical operations is low, while others may involve a large number of technical operations but may be cognitively easier.

Blain et al. (2011) introduce the Post-Editing Action (PEA), a new unit of PE effort which is a more linguistically founded way of measuring a traditional edit distance. In their approach, rather than treating each edited word as a separate action, PEAs incorporate several interrelated edit operations. For example, changing a noun propagates changes to its attributes (number, gender) which are then treated as one action. This approach has the disadvantages that it is hardly generalizable across languages, and it requires annotated corpus to train a model to classify PEAs for new texts.

A practical alternative, measuring *time* as a way of assessing post-editing effort, has only recently started to be used by researchers, although we believe this may be a more common practice in the translation industry.

Tatsumi (2009) examines the correlation between post-editing time and certain automatic metrics measuring textual differences. They find that the relationship between these two measures is not always linear, and offer some variables such as source sentence length and structure as well as specific types of dependency errors as possible explanations.

(Temnikova and Orasan, 2009; Temnikova, 2010) contrast the time translators spent fixing translations for texts produced according to a controlled language, versus translations produced using non-controlled language.

Sousa et al. (2011) compare the time spent on post-editing translations from different MT systems and on translating from scratch. The study has shown that sentences requiring less time to post-edit are more often tagged by humans as demanding low effort. It has also shown that post-editing time has good correlation with HTER for ranking both systems and segments.

Specia (2011) uses post-editing time as a way of evaluating quality estimation systems. A comparison is made between the post-editing of sentences predicted to be good and average quality sentences, showing that sentences in the first batch can be post-edited much faster.

Focusing on sub-segments, Doherty and O'Brien (2009) use an eye-tracker tool to log the fixation and gaze counts and time of translators while reading the output of an MT system. Overall translation quality was quantified on the basis of the number and the duration of fixations. Results show that fixation counts correlate well with human judgements of quality.

Following a similar approach, O'Brien (2011) measures correlations between MT automatic metrics and post-editing productivity, where productivity is measured using an eye tracker. Processing speed, average fixation time and count are found to correlate well with automatic scores for groups of segments.

Except the two latter approaches – which require eye-trackers –, to the best of our knowledge, no previous work focuses on using post-editing time as a measure of cognitive effort, and on how it correlates with technical effort. Using post-editing time for that has a number of open issues, such as the fact that it can vary significantly for different translators. In this paper we present some initial work in these two directions.

3 Materials and method

The data we have used for the experiments consists of English sentences machine translated into Spanish using eight MT systems, randomly selected from the WMT11 workshop dataset (Callison-Burch et al., 2011). The dataset includes 299 source sentences translated by two or more systems, resulting in 1484 translations. These systems were chosen based on the overall system ranking reported by WMT11: a manual evaluation had ranked the 15 participating systems in eight groups, where within each group the difference in performance was not found to be statistically significant. Within each group, we randomly picked a system: *cu-zeman*, *koc*, *online-A*, *rbmt-2*, *rbmt-4*, *rbmt-5*, *uedin*, *uow*.

	# main	# common	# all
SRC	279	20	299
SYS	8	8	8
MT	1464	20	1484
PE	1464	160	1624
MT/SRC	5.24	1	5.43

Table 1: Characteristics of the datasets: number of source sentences (SRC), systems being compared (SYS), machine translations (MT), post-edited translations (PE) and translations per source sentence (MT/SRC).

The machine translations were then edited by eight native Spanish speaking post-editors, who either were professional translators (six cases) or had some experience with post-editing (two cases). The 1484 translations were split to form two disjoint datasets (Table 1): i) a small dataset of 20 translations (one from each of 20 different sources) from randomly selected systems, and ii) a dataset made of the other 1464 translations (outputs of different systems to the remaining 279 sources - just over 5 translations per source). The first dataset (common) was edited by all the eight post-editors, that is, all of them post-edited the same 20 machine translations. The machine translations in the second dataset (main) were randomly distributed amongst the post-editors so that each of them only edited one translation for a given source sentence, and all of them edited a similar number of translations from each MT system (on average 23 per system). In sum, each post-editor edited 203 sentences (20 in common and 183 in main).

For each translation, post-editing effort indicators were logged using PET,[1] a freely available post-editing tool (Aziz et al., 2012). Among these indicators, of particular interest to our study are:

[1] http://pers-www.wlv.ac.uk/~in1676/pet/

- **TIME** the post-editing time of a sentence;
- **SPW** seconds per word, that is, the **TIME** the translator spent to post-edit the sentence divided by the length (in tokens) of the post-edited translation;
- **KEYS** the number of keystrokes pressed during the post-editing of the sentence (**PKEYS** is the subset of printable keys, that is, those that imply a visible change in the text); and
- **HTER** the standard edit distance between the original machine translation and its post-edited version (Snover et al., 2006).

Keystrokes and edit distance are natural candidates for measuring post-editing effort. To understand the usefulness of post-editing time (and its normalized version **SPW**) for this purpose, we first observed the performance of these time-based indicators at discriminating MT systems for quality. For that, we compare the system-level ranking reported by WMT11 with the rankings suggested by these indicators via Spearman's rank correlation coefficient ρ. In Table 2, the first column shows WMT11's official ranking - the numeric value is the percentage of times that the given system is better than any other system(s). The following columns show the rankings obtained by other indicators - the numeric value is the average score of each system in the `main` dataset according to that indicator. The last row shows the Spearman's rank correlation between the ranking of the gold standard (WMT11) and the ranking of each given metric. The time-based indicators, specially **TIME**, achieved a much stronger correlation with the gold standard ranking.

This initial analysis indicated that time can be a good metric to understand post-editing effort and translation quality. We then moved on to studying this metric in more detail at sentence and sub-sentence levels. More specifically, we analyse the `main` and `common` datasets in order to answer the following research questions, respectively:

- Can we characterise edits that require more cognitive effort from post-editors based on post-editing time?
- How do post-editors differ in terms of the time they spend, final translations they produce and strategies they use when post-editing?

The details on the methods used to address these two questions are given in the following sections.

3.1 Cognitive effort in post-editing

Our focus was on finding sentences that required a long time to edit and could therefore be expected to contain errors that are particularly difficult for the editor to correct. One relatively simple explanation for long editing time is sentence length (Tatsumi, 2009; Koponen, 2012). In order to target sentences where long editing time cannot be explained by sentence length alone, we chose to focus on post-editing time normalized by number of tokens in the translation. Long editing times can also be explained by the amount of editing needed in the sentence: low quality translations will naturally require more editing, but this does not necessarily mean that the edits are difficult. We thus decided to exclude cases where the sentence had undergone significant rewriting. For that, we used HTER and the observed edit operations performed as logged by PET to target sentences where relatively few changes had been made. These two indicators are different: while HTER only counts the operations that resulted in the changing of the translation, PET counts operations that were performed without necessarily changing translations, e.g, if a word is deleted and reinserted in its original form, one replacement operation is still counted.

The selection of potentially interesting examples of post-edited translations for this analysis was done with the aid of plots to visualise cases of high **SPW** and low **HTER** for each post-editor separately, to avoid any bias due to the variance in post-editing time across post-editors. For each post-editor, the four cases with the combination of longest **SPW** and lowest **HTER** were selected. A comparison set from the same post-editor with similar sentence length and similar **HTER** but short-to-average **SPW** was also selected. This was done for all post-editors, resulting in 32 cases of each type.

We then manually analysed these sentences to check if the types of errors edited differ in the two groups. Our hypothesis is that sentences with short editing times should contain more of the easy to fix errors and sentences with long edit times more of the difficult to fix errors. For error types and their level of cognitive effort, we used the 10 classes proposed in (Temnikova, 2010) with some modifica-

WMT11 ↑		**KEYS** (↓)		**HTER** (↓)		**SPW** (↓)		**TIME** (↓)	
online-A	0.72	uedin	56.29	online-A	0.229	online-A	3.06	online-A	64.48
uedin	0.64	online-A	57.04	uedin	0.242	rbmt-2	3.32	uedin	71.49
rbmt-4	0.61	rbmt-2	71.09	rbmt-2	0.281	uedin	3.33	uow	77.69
uow	0.59	rbmt-5	73.44	rbmt-5	0.291	rbmt-4	3.48	rbmt-4	78.07
rbmt-2	0.57	rbmt-4	73.81	rbmt-4	0.304	uow	3.58	rbmt-2	81.76
koc	0.56	uow	89.08	uow	0.306	rbmt-5	3.69	rbmt-5	85.20
rbmt-5	0.54	cu-zeman	94.36	koc	0.325	koc	3.84	koc	86.42
cu-zeman	0.49	koc	94.52	cu-zeman	0.331	cu-zeman	4.26	cu-zeman	100.32
Spearman's ρ		0.667		0.738		0.833		0.952	

Table 2: System-level rank correlation of each metric and WMT11's official ranking.

tions. Temnikova (2010) enriches a standard error classification (Vilar et al., 2006) for MT by ranking the error categories according to how cognitively costly she expects them to be. In addition, we hypothesise that longer edit times may involve more content words, e.g. verbs, nouns; while shorter times may involve more function words, e.g. determiners. We therefore further hypothesise that part-of-speech (POS) errors may be linked to longer edit times. Our adaptation of the classification in (Temnikova, 2010) resulted in the following error categories, from the easiest to the most difficult to fix:

0 Typographical: upper/lower case or similar orthographical edits

1 Incorrect word form

2 Incorrect style synonym: word substitutions that do not change the meaning

3 Incorrect word: divided into three cases

- **3a** Different word but same POS
- **3b** Different POS
- **3c** Untranslated source word in MT

4 Extra word

5 Missing word

6 Idiomatic expression missed

7 Wrong punctuation

8 Missing punctuation

9 Word order, word level

10 Word order, phrase level

This adaptation involved the addition of a category for orthographical edits, which is here assumed to be the easiest type. Category 3, "Incorrect word", was found to consist of different types of cases which might have cognitively different effects on the reader: an incorrect word that is the same POS as the correct one may not interfere with understanding of the sentence structure in the same way as a word that is also incorrect POS (e.g. noun instead of a verb) or an untranslated source language word. For this reason, we divided the category into three subcategories: different word but same POS, different POS, and untranslated word.

The sentences selected were lemmatised and POS tagged using the FreeLing software (Padró et al., 2010). The operation logs created by PET were used to track the changes made by the editors and then insertions, deletions and substitutions were labelled according to the error classification discussed above. Cases where the operations logged did not correspond to any changes visible in the final post-edited sentence, meaning typos and corrections made by an editor or cases where the editor revised their correction of some word or phrase several times, were not included in any error category.

3.2 Human variability in post-editing

The goal of this experiment is to analyse some aspects of the human variability in post-editing to understand whether any findings obtained using indicators from the post-editing process generalise across translators. A significant variance in segment-level post-editing time is not surprising: it is expected that different translators spend more or less time to edit the same translation, depending on their experience with the task, language-pair, text domain, etc. A variance in the final revised translations is also expected in some cases, as there is generally more than one way of expressing the source segment meaning. We were thus more interested in studying variations in the strategies used by post-editors.

We used the 20 cases from the `common` dataset that had been edited by all eight translators. These were the last translations done by all editors. We analysed the operation history logs stored by PET to observe the changes made by the editors, post-editing time, HTER and keystroke counts, including not only the overall keystroke count, but also counts on groups of specific keys pressed:

- White keystrokes: space, tab and enter
- Alphanumeric: letters (including diacritical marks) and digits
- Control: delete, backspace, combinations such as ctrl+c etc.

We hypothesise that there may be differences in the amount of "visible" typing (alphanumeric and white keys), which would reflect the individual editors' choices of how much they chose to change the translations, but also in the use of control keys, for example some editors use the arrow keys to move around in the sentence while reading and editing.

4 Results

Figure 1 shows the correlations between **TIME**, **SPW**, **HTER**, and **PKEYS** and sentence length (LEN) in the `main` data. While, as expected, absolute post-editing time grows with sentence length (5th row, 2nd col.) and number of printable keystrokes (5th row, 3rd col.), **SPW** remains fairly constant (4th row, 2nd col.). Focusing on **HTER** vs. **TIME** (5th row, 1st col.) and **HTER** vs. **PKEYS** (3rd row, 1st col.), we can see that these have most of their points concentrated at around HTER=0.5. Those regions not only contain the majority of the points (which ultimately characterises the average **TIME** and **PKEYS**), but also the highest figures for both indicators, suggesting that although HTER reflects what the final translation looks like compared to the MT, it does not reveal much about the effort required to produce that final result in terms of time and keystrokes.

4.1 Cognitive effort in post-editing

The distribution of errors in the classes adapted from (Temnikova, 2010) is shown in Figure 2. The overall pattern observed with the error distribution is that errors assumed to be most cognitively difficult (idioms, punctuation and word order errors) are indeed more common in sentences with longer editing time.

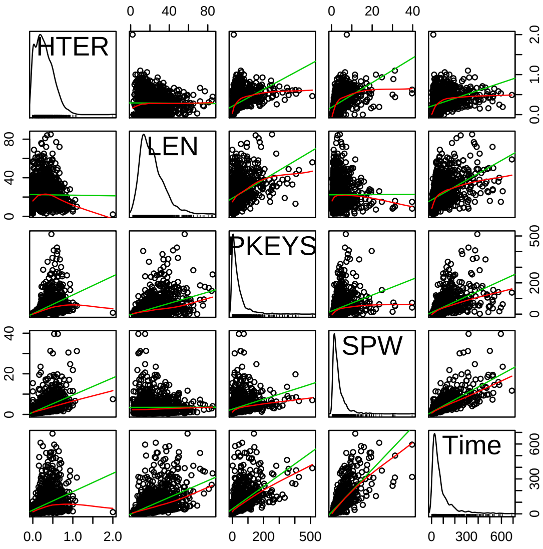


Figure 1: Scatter plot of the effort indicators: each cell in the main diagonal represents the distribution of the variable in that cell (**HTER**, **LEN**, etc.); the remaining cells correlate the variable in its column (projected on the x-axis) to the variable in its row (projected on the y-axis), and the lines are the best linear and the best polynomial fit.

For both sentences with short and long editing times, the most common errors involve category 3: "Incorrect word", with 29% in sentences with long editing time and 27% in those with short editing time. However, the distribution within the three subcategories differs: sentences with long editing time have larger proportion of the cases assumed to be more difficult, where the incorrect word has also wrong part of speech (3b) or is an untranslated source word (3c). Most of the cases in all subcategories involve nouns or verbs. Sentences with long editing times also include some 3b cases where a noun or a verb was mistranslated as an adverb. Such cases were not found in the set with short editing times.

For the sentences with short editing time, the second most common type of errors is incorrect form of a correct word (1). This is a less common type of errors in sentences with long editing times. Most

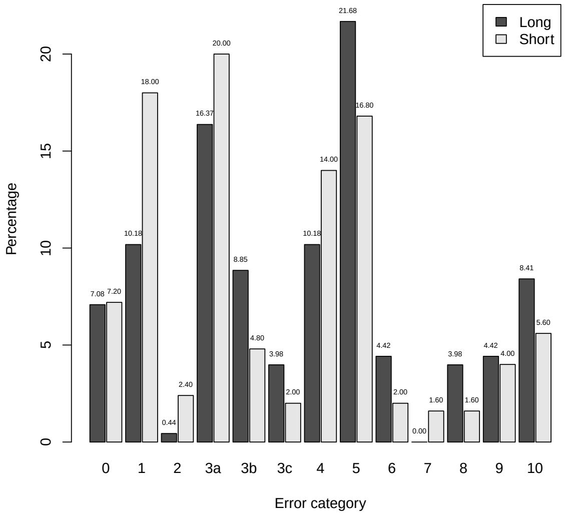


Figure 2: Comparison of error types between sentences with long and short editing times

word form errors involve verbs or determiners, but sentences with short editing times have a higher proportion (18%) of nouns with incorrect forms than those with long editing times (4%). The relative differences in proportions of word form errors between sentences with short or long editing times appears to support the ranking of these errors as “easy to fix”.

Missing words are more common in sentences with long editing times, but extra words are more common in those with short editing times. In both cases, these are mostly function words, determiners and prepositions being the largest groups. The proportion of content words is larger in sentences with long editing times with one exception: sentences with short editing times contain a few more cases of extra verbs.

Errors related to mistranslated idioms, punctuation and word order are not very common overall in either set. Mistranslated idioms involve cases where an idiomatic expression has been translated literally word for word, often changing or obscuring the meaning of the original (e.g. *(to be) at odds* translated as *en probabilidades*, literally ’at probabilities’). They are slightly more common in sentences with long editing times. Another type of literal translation can be seen where a proper noun has been erroneously translated with a common noun (e.g. as *Marca de Stanley* ’the brand of Stanley’ a for a person’s name *Stanley Brand*) or an adjective (*mala Homburg* = ’bad Homburg’ for the German place name *Bad Homburg*). Such errors were only found in the sentences with long editing time.

Cases of missing punctuation are more common in sentences with long editing times, and involve mainly missing commas. Cases of wrong punctuation (extra or replaced with other punctuation), on the other hand, were only found in the sentences with short editing times. However, at least on the surface, these few cases do not appear to be particularly critical to the understanding of the sentence: for example, substituting a comma for a semicolon or deleting an extra quotation mark. Although certain types of punctuation errors can have an effect on the meaning of a sentence by changing or obscuring the parsing of phrases, punctuation errors as a whole may not be cognitively as difficult as assumed in (Temnikova, 2010)’s classification.

Word order errors on the word level (e.g. transposition of nouns and adjectives) are about equally common in both types of sentences, but the need for reordering on the phrasal level is more common in sentences with long editing times. Furthermore, for sentences with long editing times this generally involves cases where individual words need to be reordered sometimes by long distances (and affecting the parsing of the sentence). In contrast, in sentences with short editing times about half the cases in this category involve moving groups of words into different location as whole phrases.

4.2 Human variability in post-editing

The comparison of cases where all editors post-edited the exact same machine translations show that even with the same sentence and same instructions, different editors approach the task in different ways.

Figure 3 shows the Pearson correlation coefficient for pairs of post-editors from the perspective of two different scores, namely **HTER** (bottom-left half of the figure) and **TIME** (top-right half of the figure), where darker cells indicate stronger correlation. We note that the **HTER** half is on average darker than the **TIME** half. This contributes to the hypothesis that although editors may apply a similar number of edits to the machine translation, the time they take to do it varies.

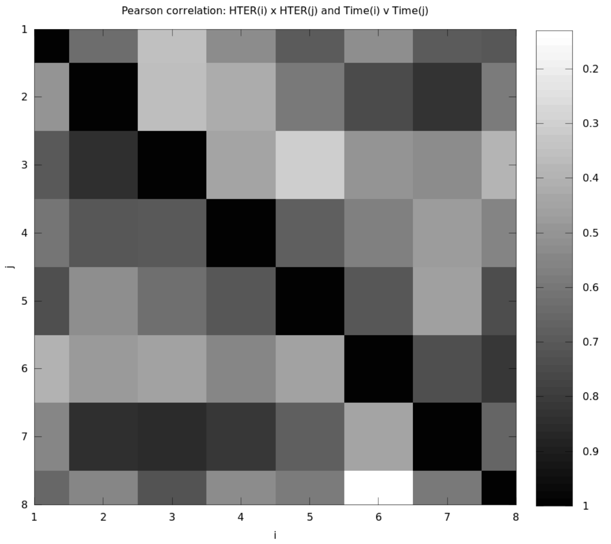


Figure 3: Each cell represents the Pearson correlation coefficient for a pair of post-editors according to **HTER** (bottom-left half) or **TIME** (bottom-right half) in the `common` dataset; darker cells indicate stronger correlation. The comparison of diagonally symmetric cells shows whether a pair of post-editors "agrees" more in terms of **HTER** or **TIME** (e.g. $\mathrm{HTER}(3,5) > \mathrm{TIME}(5,3)$; and $\mathrm{HTER}(6,8) < \mathrm{TIME}(8,6)$)

These variations may be explained by different post-editing strategies, which can be observed by comparing the different metrics: **SPW**, **HTER** and **KEYS**. Box plots for these metrics by post-editor are shown in Figure 4.

Two editors, A6 and A7, are the fastest, with the shortest editing time in 14 out of 19 sentences (in 1 case, none of the editors made any changes). On the other hand, the two slowest editors (A5 and A8) took the longest time in 11 out of the 19 cases.

In some cases, time relative to others does seem to reflect the amount of editing: the editor with the overall shortest editing times (A6) also has the lowest average **HTER**, and the two slowest editors (A5 and A8) have the highest **HTER** scores. Some differences do, however, appear: editor A4, whose editing times are third slowest of the eight editors, has in fact the second lowest **HTER**. In contrast, the second fastest editor, A7, has a considerably higher average HTER. Similarly for keystroke counts, some combine short/long editing times with low/high keystroke counts as might be expected, but despite relatively long editing times, A4 in fact uses less keystrokes on average than the two fastest editors.

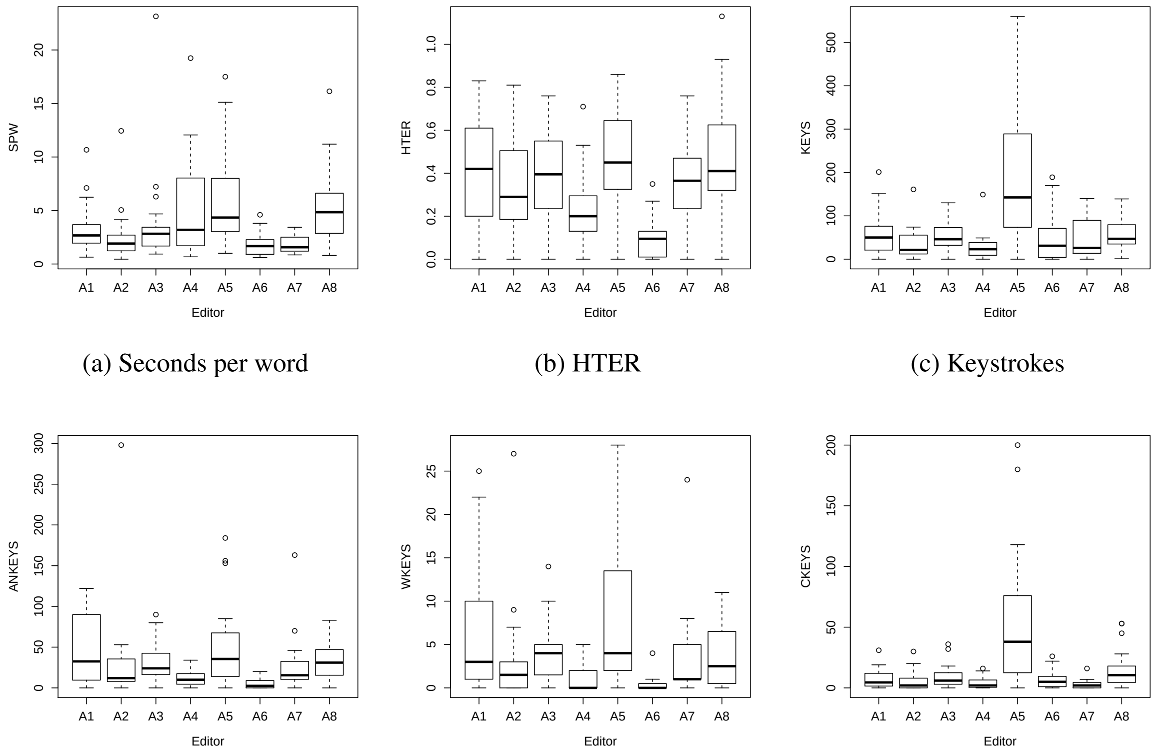
In addition to choices on how much to edit the MT sentence, some differences in post-editing times and keystrokes can also be explained by *how* the editor carries out the edits. Some editors appear make more use of the words already present in the MT as well as using cut-and-paste operations whereas others apparently prefer to type out their own version and deleting the MT words even if some parts are identical. Examples might be A7 (low **KEYS** but relatively high **HTER**) versus A1, A5 and A8 (relatively high **KEYS** and high **HTER**).

Some editors also seem to plan their edits beforehand, and edit what needs correcting only once, while others change the same passage several times or start and then delete several edits before settling on a final version. Examples may be displayed by A4 (relatively long **SPW** despite low **HTER** and low **KEYS**) versus A5 and A8 (long **SPW** combined with high **HTER** and high **KEYS**). Different editors also make the edits within the sentence in a different order, some proceeding from left to right while others move around between different parts of the sentence. Moving around inside the sentence with arrow keys may be one explanation for the very high keystroke count, and particularly high control key count by A5.

5 Conclusions

The goal of this study was to examine two questions: (i) can we characterise edits that require more cognitive effort from post-editors based on post-editing time? (ii) how do post-editors differ in terms of the time they spend, final translations they produce and strategies they use when post-editing?

The first experiment compared post-edited sentences with a long editing time to sentences with similar length and edit distance but short editing times. The errors that post-editors had corrected in these sentences were analysed according to a cognitively motivated error difficulty ranking (Temnikova, 2010), and the results suggest that the type of errors affects post-editing time. Shorter editing times seem to be associated with errors ranked cog-



(a) Seconds per word

(b) HTER

(c) Keystrokes

(d) Alphanumeric keys

(e) White keys

(f) Control keys

Figure 4: Post-editors' effort indicators in the `common` dataset.

nitively easiest, which include word form errors, synonym substitutions, and "simple" incorrect word substitutions where changing the part-of-speech is not necessary. On the other hand, substitutions involving an incorrect part-of-speech or an untranslated word, errors related to idiomatic expressions and word order, especially when reordering crosses phrase boundaries, seem to be connected with longer edit times.

These results may suggest some revisions to the assumed difficulty ranking. Sentences with short editing times in fact contained more errors labelled as extra words than sentences with long editing times. As the majority of extra/missing cases involved function words, this may indicate that extra words are not as cognitively challenging as assumed at least when they involve function words. Similarly, punctuation errors, which in turn were relatively rare in both types of sentences, showed little difference between the sentence types, and incorrect (as opposed to missing) punctuation was only found in the sentences with short editing times. Although there are certain situations where missing or incorrect punctuation could change or obscure the meaning of a sentence, perhaps not all punctuation errors need to be ranked as equally difficult.

In the second experiment, we examined post-editing effort indicators from different editors revising the same translations. Studying their variation in terms of time, edit distance and keystrokes suggests certain different editing strategies. Firstly, even with the same instructions to minimally change the machine translations, different editors make different choices about what constitutes minimal. Secondly, some editors maximize the use of MT words and cut-paste operations for reordering, while others appear to prefer writing out the whole corrected passage and then deleting MT words even when they are the same. Thirdly, some editors spend their time planning the corrections first and proceeding in order while others revise their own corrections and move around in the sentence. This could be an indication that keystrokes, while very useful as a way to understand how translators work, may not be an appropriate measure to estimate cognitive effort.

Further work is needed for truly identifying cognitively difficult errors, including analyses with larger sets, as well as different language pairs, but we believe post-editing time is a variable that should certainly be considered in analyses of this type. In addition to sentence-level post-editing time, investigating editing times related to specific operations within the sentences could provide useful information on where editors spend their time. A revised set of error categories with more detailed error types (e.g "incorrect main verb", "incorrect prepositional attachment") is also an interesting direction to help understand the cognitive load in post-editing.

Studying the strategies of different post-editors can be potentially very useful for post-editing practice. Larger scale tests with editors editing the same translations, particularly where their backgrounds and levels of experience are similar would help understand whether the variances are systematic or very specific to individual translators.

References

Wilker Aziz, Sheila C. M. Sousa, and Lucia Specia. 2012. PET: a Tool for Post-editing and Assessing Machine Translation. In *8th International Conference on Language Resources and Evaluation*, Istanbul, Turkey.

Frédéric Blain, Jean Senellart, Holger Schwenk, Mirko Plitt, and Johann Roturier. 2011. Qualitative analysis of post-editing for high quality machine translation. In *MT Summit XIII*, Xiamen, China.

C. Callison-Burch, P. Koehn, C. Monz, and O. Zaidan. 2011. Findings of the 2011 Workshop on Statistical Machine Translation. In *6th Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland.

Stephen Doherty and Sharon O'Brien. 2009. Can MT Output be Evaluated through Eye Tracking? In *MT Summit XII*, pages 214–221, Ottawa, Canada.

Maarit Koponen. 2012. Comparing human perceptions of post-editing effort with post-editing operations. In *7th Workshop on Statistical Machine Translation*, pages 181–190, Montréal, Canada.

Hans P. Krings. 2001. *Repairing texts: Empirical investigations of machine translation post-editing process*. The Kent State University Press, Kent, OH.

Sharon O'Brien. 2011. Towards predicting post-editing productivity. *Machine Translation*, 25(3):197–215, September.

Joseph Olive, Caitlin Christianson, and John McCary. 2011. *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*. Springer.

Lluís Padró, Miquel Collado, Samuel Reese, Marina Lloberes, and Irene Castellón. 2010. FreeLing 2.1: Five Years of Open-Source Language Processing Tools. In *7th International Conference on Language Resources and Evaluation*, pages 3485–3490.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *7th Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts.

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2010. TER-Plus: Paraphrase, Semantic, and Alignment Enhancements to Translation Edit Rate. *Machine Translation*, 22(2-3):117–127.

Sheila C. M. Sousa, Wilker Aziz, and Lucia Specia. 2011. Assessing the post-editing effort for automatic and semi-automatic translations of DVD subtitles. In *Recent Advances in Natural Language Processing Conference*, Hissar, Bulgaria.

Lucia Specia. 2011. Exploiting objective annotations for measuring translation post-editing effort. In *15th Conference of the European Association for Machine Translation*, pages 73–80, Leuven, Belgium.

Midori Tatsumi. 2009. Correlation between Automatic Evaluation Metric Scores, Post-Editing Speed, and Some Other Factors. In *MT Summit XII*, pages 332–33.

Irina Temnikova and Constantin Orasan. 2009. Post-editing Experiments with MT for a Controlled Language. In *International Symposium on Data and Sense Mining, Machine Translation and Controlled Languages*, Besançon, France.

Irina Temnikova. 2010. A Cognitive Evaluation Approach for a Controlled Language Post-Editing Experiment. In *7th International Conference on Language Resources and Evaluation*, Valletta, Malta.

David Vilar, Jia Xu, Luis Fernando D'Haro, and Hermann Ney. 2006. Error Analysis of Machine Translation Output. In *5th International Conference on Language Resources and Evaluation*, pages 697–702.

Average Pause Ratio as an Indicator of Cognitive Effort in Post-Editing: A Case Study

Isabel Lacruz
Kent State University
Kent, OH 44242, U.S.A.
ilacruz@kent.edu

Gregory M. Shreve
Kent State University
Kent, OH 44242, U.S.A.
gshreve@neo.rr.com

Erik Angelone
Kent State University
Kent, OH 44242, U.S.A.
eangelon@kent.edu

Abstract

Pauses are known to be good indicators of cognitive demand in monolingual language production and in translation. However, a previous effort by O'Brien (2006) to establish an analogous relationship in post-editing did not produce the expected result. In this case study, we introduce a metric for pause activity, the *average pause ratio*, which is sensitive to both the number and duration of pauses. We measured cognitive effort in a segment by counting the number of complete editing events. We found that the average pause ratio was higher for less cognitively demanding segments than for more cognitively demanding segments. Moreover, this effect became more pronounced as the minimum threshold for pause length was shortened.

1 Introduction

A fundamental objective of machine translation is to reliably produce high quality translations. Much progress has been made in automatically rating the quality of MT production (see O'Brien, 2011 for a discussion), and, over time, incorporating ratings into MT systems could reduce the need for post-editing. However, post-editing remains a significant activity that involves considerable human effort. A better understanding of the factors that contribute to post-editing effort is important, since the level of effort expended by the post-editor is closely tied to productivity.

Our understanding of post-editing effort is still far from complete, although there is a growing body of research on its nature. An important early contribution was the work of Krings (2001). He classified post-editing effort into three distinct categories: temporal (time spent), cognitive (mental processing), and technical (physical action). In his view, temporal effort results from a combination of cognitive and technical effort. Temporal and technical effort can be measured accurately with the help of modern technology.

Total post-editing time is the most basic measurement of temporal effort, but researchers have also used keystroke logging and eye-tracking to measure pause times or gaze duration (e.g., Krings 2001; O'Brien 2004; O'Brien 2005; O'Brien 2006; Dragsted and Hansen 2009; Carl et al. 2011).

Technical effort is the work involved in the keyboarding and mouse actions needed to make changes to the MT output. It can be measured by using logging technology to count the various possible actions, including insertion, deletion, cutting, and pasting (e.g. Krings 2001; O'Brien 2004; O'Brien 2005; O'Brien 2006). Aikawa et al. (2007) used a character-based metric to gauge the variation between MT and post-edited texts. Another approach is to use automatic metrics to measure the distance between the MT text and its final post-edited version (e.g. Tatsumi 2009; Temnikova 2010; Koponen 2012).

On the other hand, the mental processing involved in cognitive effort cannot be measured so directly. Researchers have investigated several approaches to measuring post-editing effort and the factors that contribute to it. These include rating the translatability of the source text (e.g. O'Brien

2006), rating post-editing difficulty in the MT text through think-aloud protocols (Krings 2001), choice-network analysis (O'Brien 2005; O'Brien 2006), ranking classifications of error difficulties (e.g. O'Brien 2006; Temnikova 2010; Koponen 2012), or effort ratings (e.g. Specia et al. 2009).

In this paper, we propose a new approach to measuring cognitive effort in post-editing. We classify post-edited segments as having required more or less cognitive effort on the part of the post-editor based on a metric that counts the number of *complete editing events*. In many circumstances, collections of individual editing actions can be considered to naturally form part of the same overall action, which is what we label as a complete editing event. For example, the insertion of a word by typing three characters separated by pauses is classified as a single complete editing event, not three separate editing events. This highlights the possible role of clusters of short pauses as indicators of cognitive effort. It suggests that total pause time in a segment may not by itself be an accurate indicator of cognitive effort in post-editing. This prompts us to introduce a new pause metric for segments and to investigate how it relates to our technical measure of cognitive effort.

Pauses, measured by keystroke logging or by eye tracking data on fixations and gaze duration, are known to be good indicators of cognitive demand in monolingual language production (e.g., Schilperoord 1996) and in translation and interpreting (e.g. Krings 2001, Dragsted and Hansen 2008, Shreve, Lacruz, and Angelone. 2011; Timarová, Dragsted, and Hansen 2011). It is therefore natural to expect pauses in post-editing to be indicators of cognitive demand. Surprisingly, previous post-editing studies did not find significant evidence for a relationship between pauses and cognitive demand (O'Brien, 2006). O'Brien compared the *pause ratio* (total time in pause divided by total time in segment) for machine translated segments where the source text had different concentrations of negative translatability indicators. These are linguistic features, such as passive voice, long noun phrases, or ungrammatical constructs, which are known to be problematic for machine translation. O'Brien predicted that segments with one or more negative translatability indicators would result in greater cognitive demands on the post-editor. She hypothesized that increased cognitive load should correspond to increased pause activity, as measured by the pause ratio, which she computed using pauses with a duration of at least one second. However, she subsequently found no significant difference in pause ratios for more or less cognitively demanding segments.

Nevertheless, the research cited previously in monolingual language production and in translation and interpreting provides strong evidence that there should be a relationship between cognitive load and pause activity in any environment involving reading and language production, including post-editing. This suggests that in O'Brien's study either the measurement of cognitive load or the metric for describing pause activity were insufficiently sensitive to reveal the expected effect. To follow up on O'Brien's initial investigation of this area and to dig deeper into these issues we conducted a case study in which we changed both the measurement of cognitive load and the metric for the pause activity.

O'Brien (2006) predicted that cognitive effort in post-editing would depend on features of the source text that would make it more or less difficult to translate by machine. This assumes the MT will be harder to post-edit when the source text has negative translatability indicators than when it does not. However, this is an indirect measure of cognitive effort in post-editing. To obtain a more direct measure, we focused on actual post-editing activity. Each post-editor is likely to experience different challenges, depending on his or her experience. Accordingly, we assessed the cognitive demand imposed by each segment using a measure of technical effort. We counted the number of complete editing events. We used this measure of technical effort to classify the post-edited segments into two categories (more or less cognitively demanding) depending on whether there were more or fewer complete editing events in the segment under consideration.

Pause activity can manifest itself in a variety of ways that cannot be discriminated by a measure based on total pause time in the segment. Pauses are of variable length, and a large number of short pauses will likely indicate a different cognitive processing/effort pattern than a single pause of thesame overall duration. Such differences can be captured to some extent by using the *average pause ratio*, which is computed for each segment as the average time per pause in the segment divided by the average time per word in the segment. We used these alternative assessments of cognitive load and pause activity to search for the expected relationship between cognitive load and pause activity in post-editing.

2 Method

The participant in the case study (L1 English and L2 Spanish) was a professional translator with 25 years experience as a freelance translator, 13 years of classroom experience in editing translations for pedagogical purposes, and four years of experience as a literary translation journal editor. He had no previous experience with post-editing machine translated text and no experience with software manuals.

The volunteer participant was seated in a quiet office and was asked to post-edit a MT text to his satisfaction. The text was part of a software instruction manual in English and that had been machine translated into Spanish using a phrase-based Moses system. No time constraint was imposed. The text was divided into segments roughly corresponding to sentences. Segment length ranged from 5 to 38 words with a mean of 19.4 words (median 23 words.). There were a total of 15 segments. The materials are included in Appendix A.

The Translators Workbench program from SDL Trados was used to present segments one by one on a computer screen, with the source text segment appearing at the top of the screen and the TM-proposed MT segment underneath. The participant was asked to post-edit the MT segments, and a keystroke log was recorded using the Inputlog keystroke logger.

3 Rationale

The post-editing of a segment can be broken down into the following steps:

- Reading of the presented source and target text segments
- *Problem recognition* based on a comparison of the source text segment with the target text segment
- Decision to act (accept, revise, or reject and re-write) the target text segment based on problem recognition results
- *Solution proposal* for identified translation problems if a decision is made to revise or re-write
- Post-editing action based on a selected solution proposal
- *Solution evaluation* of post-edited segment result
  - If not acceptable, revise or re-write again
  - If acceptable, continue to the next segment.

These steps are based on Angelone's (2010) three-stage behavioral model for uncertainty management in translation. The first stage, reading, invokes "the ability to extract visual information from the page and comprehend the meaning of the text" (Rayner and Pollatsek, 1989) and of the MT text. The stages identified by Angelone were *problem recognition*, *solution proposal*, and *solution evaluation*. These three stages are the most likely loci for cognitive effort in the active production part of post-editing, and it is natural to expect this effort to be observable in the pause data.

Indeed, we inferred very different pause patterns in the different steps of the post-editing process, based on keystroke observations. In particular, there were distinctive distributions of long and short pauses at each stage. For the purposes of the discussion below, short pauses are those that last for less than two seconds, while long pauses last at least five seconds. We frequently observed clusters of long pauses during the reflective stages of reading, problem recognition, and solution proposal, stages that place high cognitive demand on the post-editor. Final decision to act was often preceded by a single short pause. It was also notable that concentrated clusters of short pauses tended to accompany complex post-editing action; these clusters appear to be additional indicators of high cognitive demand. Finally, during the solution evaluation phase we again observed clusters of long pauses, which are again associated with high cognitive demand.

These observed patterns of long and short pauses appear to correspond in different ways to the cognitive effort expected at each stage. In particular, high cognitive load appears to be associated with both long pauses and clusters of short pauses. The *pause ratio* (total time in pause divided by total time in segment) does not take different patterns of pause behavior into account. In particular, it is not sensitive to the existence of clusters of short pauses. This prompted us to introduce the *average pause ratio* (average time per pause in a segment divided by average time per word in the segment) as a measure that is sensitive to different distributions of long and short pauses.

We consider illustrative examples to highlight the distinction between pause ratio and average pause ratio for segments. Take as a baseline a twenty-word segment that takes 80 seconds to post-edit, including a total time of 40 seconds in several pauses of varying duration. Regardless of the number and duration of individual pauses, the pause ratio for such a segment will always be 40/80 = 0.5. Now consider three distinct pause patterns outlines in Table 1 below, each consistent with the baseline description.

Case	Number of 1 sec pauses	Number of 20 sec pauses	Total pauses in segmen t	Pause time in segment (secs)
A	0	2	2	40
B	20	1	21	40
C	40	0	40	40

Table 1: Examples of segments with varying pause distributions, but the same overall time in pause

To compute the average pause ratio in any of these scenarios, we need to compute the average time per pause and the average time per word. See Table 2 below. The *average time per word* is the same in all three scenarios. It is:

(total time in segment)/(# of words in segment).

Thus, the average time per word is 80/20 sec = 4.0 sec.

However, the *average time per pause*, computed as:

(total time in pause)/(# of pauses in segment),

is different in each of the three scenarios. This is because the number of pauses varies from scenario to scenario, due to the different patterns of individual pause durations, while the total time in pause (40 sec) is the same in each scenario.

Case	Average time per pause (sec)	Average time per word (sec)	Average pause ratio
A	20	4	5.0
B	1.9	4	0.48
C	1	4	0.25

Table 2: Examples of average pause ratios

To summarize, these examples serve to illustrate the sensitivity of the average pause ratio to different pause patterns that do not affect the pause ratio.

It is also worth noting the effect of extending the total pause time, For example, if scenario A were modified so that there were four 20 second pauses instead of two, the average time per pause would still be 20 sec, but the total time in segment would increase to 120 sec, causing the average time per word to change to 120/20 sec = 6 sec. As a result, the average pause ratio would change from 5.0 to 20/6 = 3.3. In this situation, the pause ratio would also change - from 0.5 to 0.67.

4 Results

Intuitively, as the number of complete editing events rises, the level of overall cognitive demand experienced by the post-editor should increase. We classified the post-edited segments as *more cognitively demanding* when there were 4 or more complete editing events and *less cognitively demanding* when there were 2 or fewer complete editing events.

In order to create a clear separation between the two categories, we chose to remove from analysis the two segments with 3 complete editing events. (However, we note that the results we obtain would not have been significantly different if we had included the segments with 3 complete editing events in the less cognitively demanding group.) The choice of how to separate the more demanding group of segments from the less demanding group was based on clear breaks in the distribution of complete editing events around the median of 4.

It is important to emphasize that a large scale experimental study involving several individuals is needed to scientifically explore the way in which cognitive effort is related to the number or concentration of complete editing events.

Of the 13 segments analyzed, 8 were more cognitively demanding and 5 were less cognitively demanding. Data about the distribution of edits in each category is given in Table 3 below.

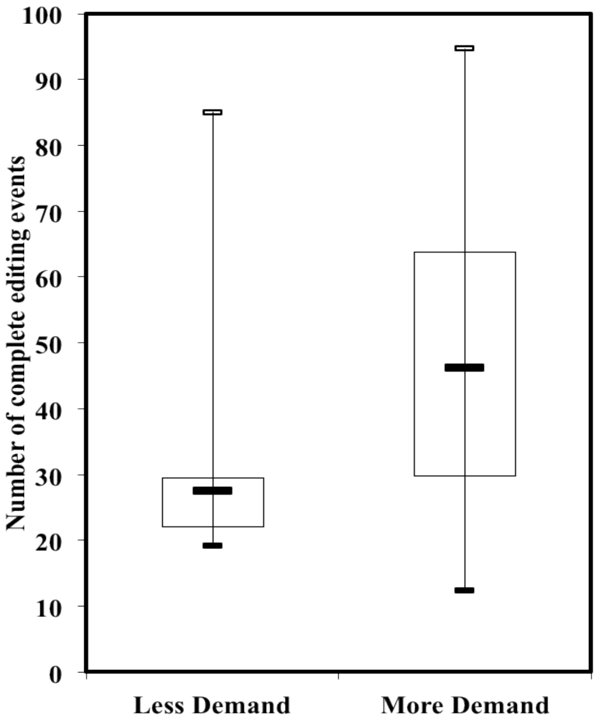


Figure 1: Boxplots of the distributions of complete editing events for more and less cognitively demanding segments

The length distribution of the more cognitively demanding segments (mean 19.0 words) was comparable to that for the less cognitively demanding segments (mean 17.2 words). See Figure 2 below.

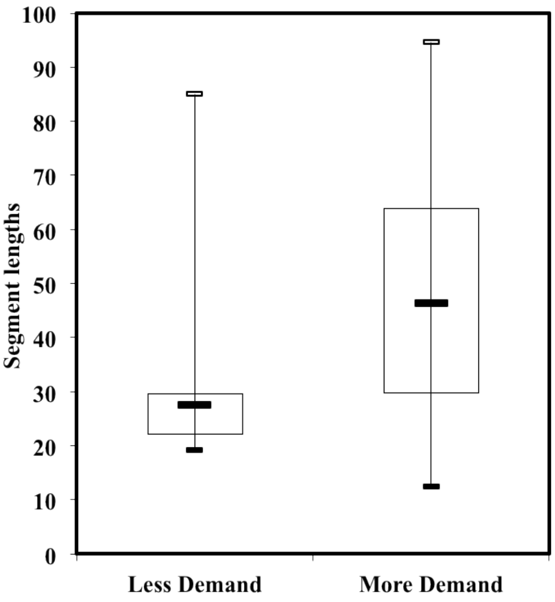


Figure 2: Boxplots the distributions of segment lengths for more and less cognitively demanding segments

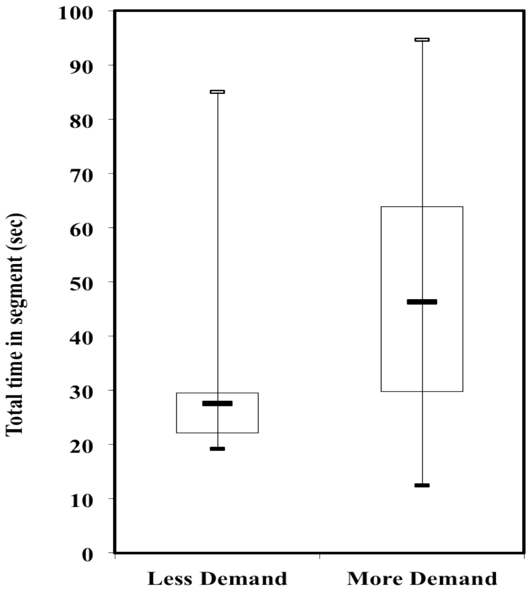


Figure 3: Boxplots of the distributions of total time in segment (sec) for more and less cognitively demanding segments

For more cognitively demanding segments total post-editing time (mean 111.5 sec) and total time in pause (48.2 sec) was longer than for less cognitively demanding segments (62.0 sec and 36.7 sec, respectively.) See Figures 3 and 4.

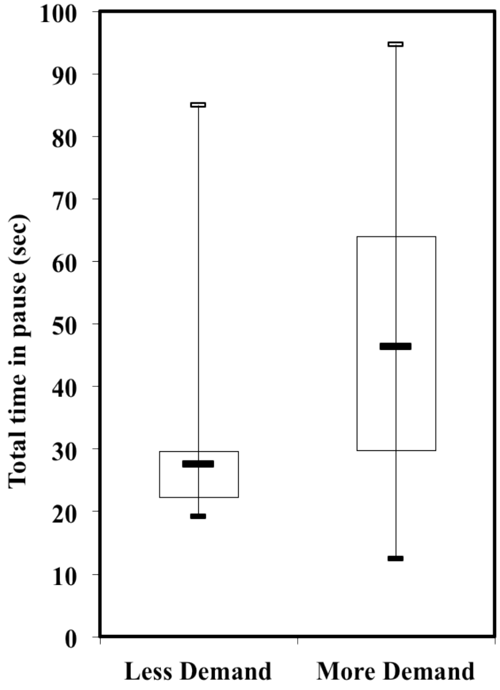


Figure 4: : Boxplots of the distributions of total time in segment (sec) for more and less cognitively demanding segments

We predicted that more cognitively demanding segments would have many short pauses associated with the monitoring of the higher number of post-editing actions. The predominance of short pauses should result in a low average pause ratio. On the other hand, in less cognitively demanding segments most of the effort would be in reading comprehension, problem recognition, and solution evaluation, where we typically found clusters of long pauses. The predominance of long pauses should result in a high average pause ratio.

In more cognitively demanding segments, pause ratio for pauses longer than 1 second was 0.42, while for less cognitively demanding segments it was 0.51. (See Figure 5.) A one-tailed independent samples t-test showed the pause ratio for less demanding segments was not significantly greater than for more demanding segments, t(11) = 1.16, p = .13.

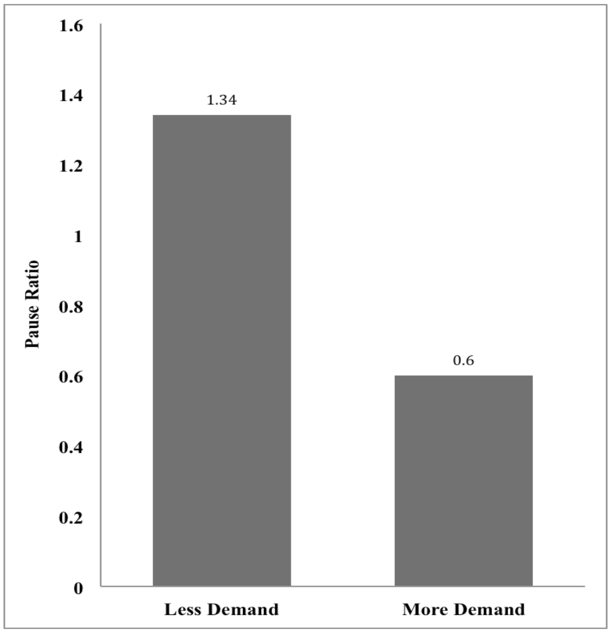


Figure 5: Pause Ratio for More and Less Cognitively Demanding Segments

O'Brien (2006) found that pause ratio in post-editing was not changed when her indirect measurement of cognitive load (based on features of the source text) was increased. The present result indicates that pause ratio is also unchanged when our more direct measurement of cognitive load (based on post-editor behavior) is increased.

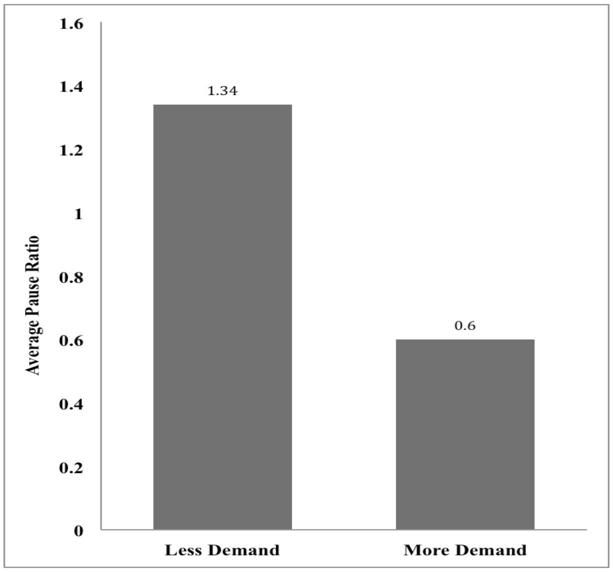
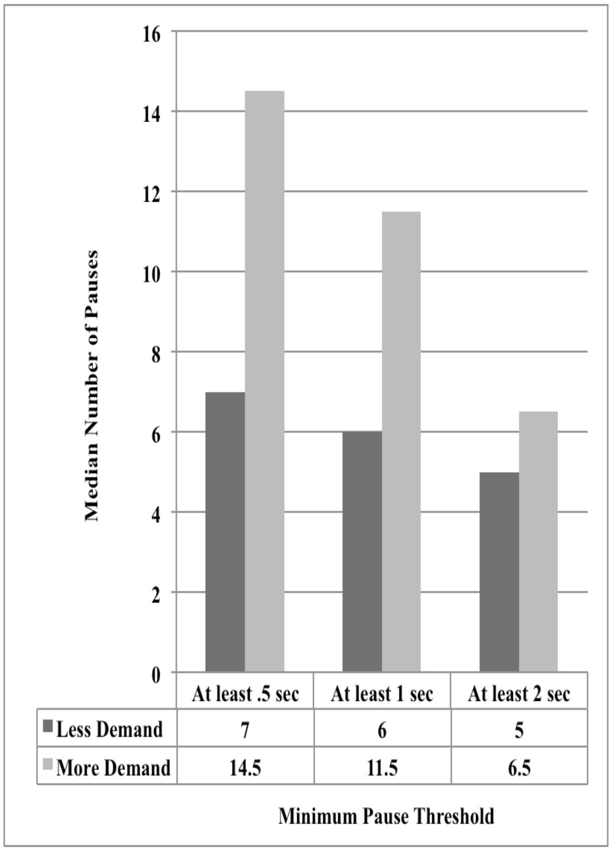


Figure 6: Average Pause Ratio for More and Less Cognitively Demanding Segments

On the other hand, in more cognitively demanding segments average pause ratio for pauses longer than 1 second was .60, while it was 1.34 for less cognitively demanding segments. (See Figure 6.) A one-tailed independent samples t-test showed the observed average pause ratio for less demanding segments was significantly higher than that for more demanding segments, t(11) = 2.63, p = .01. This indicates that average pause ratio decreases as predicted when our output measurement of cognitive load is increased.

We also computed average pause ratios for three different minimum pause durations: half-second, one second, and two seconds. As this lower threshold decreases, more cognitively demanding segments should gain more (short) pauses than less cognitively demanding segments. Consequently, although the average pause ratio for both types of segment should decrease, the predicted variation in average pause ratio should become more marked.



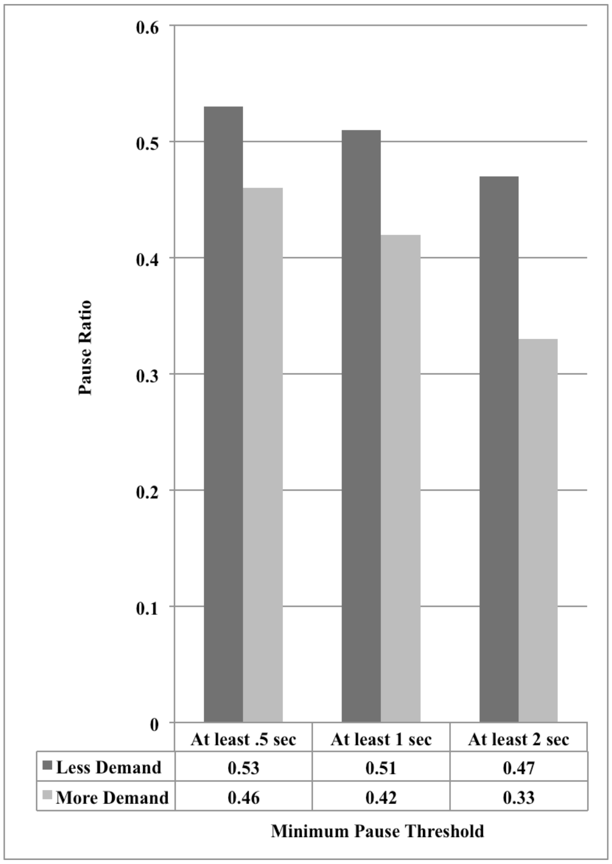
	At least .5 sec	At least 1 sec	At least 2 sec
■ Less Demand	7	6	5
■ More Demand	14.5	11.5	6.5

Figure 7: Median Number of Pauses at Different Minimum Pause Thresholds

As predicted, the number of pauses in more cognitively demanding segments increased faster than the number of pauses in less cognitively demanding segments as the minimum pause length was reduced. See Figure 7.

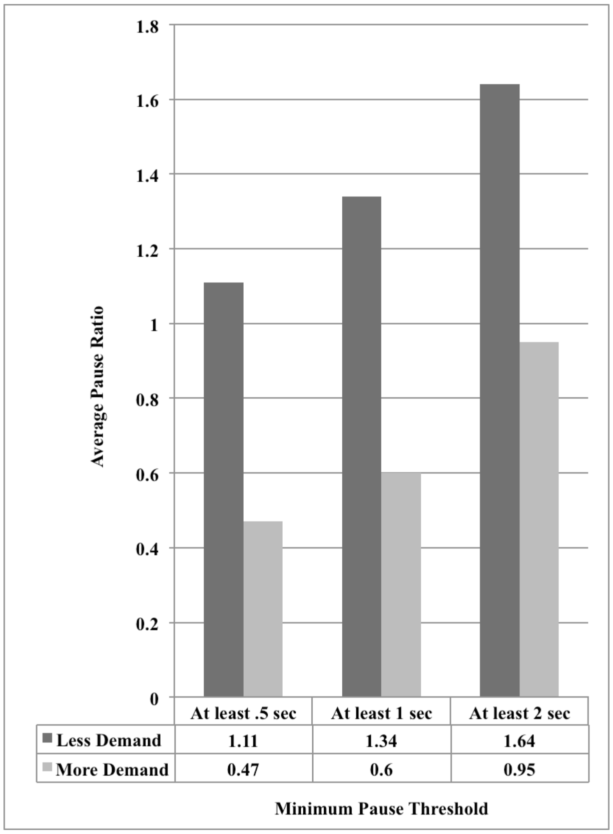
Moreover, the results we found for the 1-second minimum pause threshold continued to hold for other threshold levels. The pause ratios corresponding to each minimum pause threshold level were not significantly different for more and less cognitively demanding segments. (See Figure 8.)

However, as predicted, more cognitively demanding segments had significantly smaller average pause ratio than less cognitively demanding segments, and this effect became proportionally more marked as the lower threshold for pause time was reduced. (See Figure 9.)



	At least .5 sec	At least 1 sec	At least 2 sec
■ Less Demand	0.53	0.51	0.47
■ More Demand	0.46	0.42	0.33

Figure 8: Means of Pause Ratios at Different Minimum Pause Thresholds



	At least .5 sec	At least 1 sec	At least 2 sec
■ Less Demand	1.11	1.34	1.64
■ More Demand	0.47	0.6	0.95

Figure 9: Means of Average Pause Ratios at Different Minimum Pause Thresholds

5 Conclusions and future directions

The main contribution of this paper is the identification of the average pause ratio metric as a potentially valid measure of cognitive demand. However, it is important to emphasize that our results are based on a case study of post-editing behavior in a single individual using a small number of MT segments. Our findings cannot be generalized to other situations without careful experimental replication involving several individuals and a larger segment pool.

We found a relationship between cognitive demand and average pause ratio: for more cognitively demanding segments the average pause ratio was smaller than for less cognitively demanding segments. This difference was significant for pauses longer than .5, 1, and 2 seconds.

Furthermore, we found that as the pause length threshold decreased the proportional difference between more and less cognitively demanding segments became greater. These effects are consistent with our observation that post-editing actions are often accompanied by a proliferation of short pauses.

Cognitive demand was measured by counting the number of complete post-editing events in the post-edited text. It is important to investigate the impact of individual differences on this measure. A subsequent goal would be to predict cognitive demand on the post-editor, not from the actions of the post-editor, but from characteristics of the target text itself - and eventually from characteristics of the source text.

A systematic investigation of the patterns of pauses we observed in this case study has the potential to provide a means to reliably delineate the different stages of the post-editing process through pause patterns. This could be done empirically, for example by varying error type and error location in target text segments.

The scope of the effect of error type on cognitive demand should also be investigated. Some MT errors result in significant loss of meaning, while other errors have a more superficial impact. Is there a relationship between the type of MT error and the pattern of pauses? When errors cause significant loss of meaning, is it easier for the post-editor to re-write rather than to post-edit?

References

Takako Aikawa, Lee Schwartz, Ronit King, Mo Corston-Oliver, and Carmen Lozano. 2007. Impact of Controlled Language on Translation Quality and Post-Editing in a Statistical Machine Translation Environment. Proceedings of the MT Summit XI, (pp. 1-7). Copenhagen, Denmark

Rui A. Alves, Castro, Sao L. Castro, and Thierri Olive. 2008. Execution and Pauses in Writing Narratives: Processing Time, Cognitive Effort and Typing Skill. International Journal of Psychology, 43(6), 969-979.

Erik Angelone. 2010. Uncertainty, Uncertainty Management, and Metacognitive Problem Solving in the

Translation Task. In Gregory M. Shreve and Erik Angelone (Eds.). *Translation and Cognition*, (pp. 17-40). Amsterdam/Philadelphia: John Benjamins.

Michael Carl, Barbara Dragsted, Jakob Elming, Daniel Hardt, and Arnt Lykke Jakobsen. 2011. The Process of Post-Editing: A Pilot Study. Proceedings of the 8th International NLPSC workshop. Special theme: Human machine interaction in translation. Copenhagen Studies in Language, 412. Frederiksberg: Samfundsliteratur.

Barbara Dragsted and Inge Gorm Hansen. 2008. Comprehension and Production in Translation: a Pilot Study on Segmentation and the Coordination of Reading and Writing Processes. In Susanne Göpferich, Arnt Lykke Jakobsen, and Inger M. Mees (Eds.), *Looking at Eyes* (pp. 9–30). Copenhagen Studies in Language 36. Copenhagen: Samsfundslitteratur.

Barbara Dragsted and Inge Gorm Hansen. 2009. Exploring Translation and Interpreting Hybrids. The Case of Sight Translation. Meta: Translators' Journal, 54(3), 588-604.

Maarit Koponnen. 2012. Comparing Human Perceptions of Post-Editing Effort with Post-Editing Operations. Proceedings of the Seventh Workshop on Statistical Machine Translation (pp. 181–190). Montreal (Canada).

Hans P. Krings. 2001. Repairing Texts: Empirical Investigations of Machine Translation Post-Editing Processes. Geoffrey S. Koby (Ed.). Kent, Ohio: Kent State University Press.

Sharon O'Brien. 2004. Machine Translatability and Post-Editing effort: How do they relate? Translating and the Computer, 26.

Sharon O'Brien. 2005. Methdologies for Measuring the Correlations Between Post-Editing Effort and Machine Text Translatability. Machine Translation, 19(1): 37-58.

Sharon O'Brien. 2006. Pauses as Indicators of Cognitive Effort in Post-editing Machine Transl- ation Output. Across Languages and Cultures, 7(1), 1-21.

Sharon O'Brien. 2011. Towards Predicting Post-Editing Productivity. Machine Translation, 25, 197-215.

Keith Rayner and Alexander Pollatsek. 1989. The Psychology of Reading. Hillsdale, New Jersey: Laurence Erlbaum Associates.

Joost Schilperoord. 1996. It's About Time: Temporal Aspects of Cognitive Processes in Text Production. Amsterdam: Rodopi.

Gregory M. Shreve, Isabel Lacruz, and Erik Angelone. 2011. Sight translation and Speech Disfluency: Performance Analysis as a Window to Cognitive Translation Processes. In Cecilia Alvstad, Adelina Hild, and Elisabet Tiselius (Eds.), *Methods and Strategies of Process Research* (pp. 121–146). Amsterdam/ Philadelphia: John Benjamins.

Lucia Specia, Nicola Cancedda, Marco Dymetman, Marco Turchi, and Nello Cristianini. 2009. Estimating the Sentence-Level Quality of Machine Translation Systems. Proceedings of the 13th Annual Conference of the EAMT (pp. 28-35). Barcelona, Spain.

Midori Tatsumi. 2009. Correlation Between Automatic Correlation Evaluation Scores, Post-Editing Speed, and Some Other Factors. Proceedings of the MT Summit XII (pp. 386-393). Ottawa, Canada.

Irina Temnikova. 2010. A Cognitive Evaluation Approach for a Controlled Language Post-Editing Experiment. International Conference Language Resources and Evaluation (LREC2010). Valletta, Malta, May 17-23.

Sárka Timarová, Barbara Dragsted, and Inge Gorm Hansen. 2011. Time Lag in Translation and Interpreting. In Cecilia Alvstad, Adelina Hild, and Elisabet Tiselius (Eds.), *Methods and Strategies of Process Research* (pp. 121–146). Amsterdam/ Philadelphia: John Benjamins.

Appendix A: Sample Materials

Less cognitively demanding segments (0-2 complete editing events):

ST: Google Docs has a revision history pane that allows you to view at a glance all changes made to a doc by each collaborator.
MT: Google Docs tiene un panel Historial de revisiones que le permite ver de un vistazo para todos los cambios realizados en un documento por cada colaborador.
PE: Google Docs tiene un panel Historial de revisiones que le permite ver de un vistazo todos

los cambios realizados en un documento por cada colaborador.

ST: Click a time stamp in the right column to see what changes were made at a given time or use the arrow keys to scan through many revisions quickly.
MT: Haga clic en una marca de fecha y hora en la columna de la derecha para ver qué cambios se han realizado en un momento determinado o utilice las teclas de flecha para escanear a través de numerosas revisiones rápidamente.
PE: Haga clic en una marca de fecha y hora en la columna de la derecha para ver qué cambios se han realizado en un momento determinado o utilice las teclas de flecha para revisar numerosas revisiones de forma rápida.

ST: If you'd like to revert to the version you're currently viewing, click Restore this revision.
MT: Si desea revertir a la versión que está viendo, haga clic en Restaurar esta revisión.
PE: Si Ud. desea revertir a la versión que está viendo, haga clic en Restaurar esta revisión.

More cognitively demanding segments (3 or more complete editing events):

ST: For example, James, whose edits show in orange text, deleted and added text while bmichael, whose show in green text, removed a paragraph and added a comment.
MT: Por ejemplo, Juan, cuyos cambios se muestran en naranja texto, elimina y se agrega texto al bmichael, cuyo texto aparezcan en verde, elimina un párrafo y se agrega un comentario.
PE: Por ejemplo, Juan, cuyos cambios se muestran en texto anaranjado, elimino' y se agrego' texto mientras que bmichael, cuyo texto aparece en verde, elimino' un párrafo y agrego' un comentario.

ST: Google spreadsheets sometimes trims down your revisions over time to save storage.
MT: Google hojas de cálculo a veces recorta hacia abajo las revisiones a lo largo del tiempo para guardar su almacenamiento.
PE: Las hojas de cálculo Google a veces reduce las revisiones a lo largo del tiempo para reducir la cantidad de almacenamiento necesaria.

ST: Note: Restoring your document to a previous version does not eliminate any versions of your document; rather this version moves to the top of your revision history, maintaining all previous versions of your document, including the current version.
MT: Note: Nota: restaurar el documento a una versión anterior no se eliminan todas las versiones del documento. En lugar de esta versión se mueve a la parte superior de su historial de revisiones, mantener todas las versiones anterior del documento, incluida la versión actual.
PE: No'tese: El restaurar el documento a una versión anterior no elimina todas las versiones del documento. En cambio esta versión se mueve al primer lugar de su historial de revisiones, manteniendo todas las versiones anteriores del documento, inclusive la versión actual.

ST: Visit the Revision Pruning help article to learn more about this process.
MT: Visite el artículo de ayuda de eliminación de revisión para obtener más información sobre este proceso.
PE: Para informarse ma's sobre este proceso, consulte el artículo de soporte sobre los Recortes de las revisiones.

Segment excluded from analysis (3 complete editing events):

ST: If you're working in Google spreadsheets, and your document is either large or you created it a long time ago, your revisions may be pruned.
MT: Si está trabajando en Google, hojas de cálculo, y el documento es grande o lo creó hace mucho tiempo, las revisiones se pueden cortar.
PE: Si Ud. está trabajando en las hojas de cálculo Google, y el documento o es grande o lo creó hace mucho tiempo, las revisiones se pueden recortar.

Reliably Assessing the Quality of Post-edited Translation Based on Formalized Structured Translation Specifications

Alan K. Melby
Brigham Young University
Department of Linguistics and English Language
akmtrg@byu.edu

Jason Housley
Brigham Young University
Translation Research Group
housleyjk@gmail.com

Paul J. Fields
Brigham Young University
Continuing Education
pjfphd@byu.net

Emily Tuioti
Brigham Young University
Department of Linguistics and English Language
emily.tuioti@gmail.com

Abstract

Post-editing of machine translation has become more common in recent years. This has created the need for a formal method of assessing the performance of post-editors in terms of whether they are able to produce post-edited target texts that follow project specifications. This paper proposes the use of formalized structured translation specifications (FSTS) as a basis for post-editor assessment. To determine if potential evaluators are able to reliably assess the quality of post-edited translations, an experiment used texts representing the work of five fictional post-editors. Two software applications were developed to facilitate the assessment: the Ruqual Specifications Writer, which aids in establishing post-editing project specifications; and Ruqual Rubric Viewer, which provides a graphical user interface for constructing a rubric in a machine-readable format. Seventeen non-experts rated the translation quality of each simulated post-edited text. Intraclass correlation analysis showed evidence that the evaluators were highly reliable in evaluating the performance of the post-editors. Thus, we assert that using FSTS specifications applied through the Ruqual software tools provides a useful basis for evaluating the quality of post-edited texts.

1 Research Question

The progression of globalization has produced an ever increasing demand for materials to be translated. Moreover, there are now an insufficient number of highly skilled translators to handle the total demand for translation services. In other words, the world has moved beyond the point when there were more than enough translators to meet the demand (Hutchins, 2007), into an era where the need for translation has made the use of machine translation (MT) widespread.

One practical application of MT that allows it to be applied to a variety of situations is post-editing, that is, the addition of a human editor to correct raw MT output to meet a set of requirements that the MT system would not be able to fully meet on its own. Post-editing presents a set of problems and challenges, not the least of which is assessing how well human editors can perform as post-editors. If the target-text production performance of human post-editors cannot be assessed reliably, then other measures, such as post-editing speed, are meaningless. Therefore, this study was focused on reliability.

As a step toward a general answer, we asked a specific research question: How reliably can non-expert human evaluators assess the quality of post-edited machine translations given three conditions:

1. The initial English target text was generated by a free and publically available machine translation system;
2. The source text was a medium difficulty (Interagency Language Roundtable (ILR) Level 2) document in Japanese; and
3. The evaluators assessing the performance of the post-editors were given a rubric based on a set of structured translation specifications?

Here performance is defined as “the ability to produce a target text that meets agreed-upon project specifications.” Therefore, a quality target text is one that meets the specifications. Note that this is a functional view of quality, not an absolute view of quality, which would require a target text to be completely accurate and perfectly fluent, regardless of audience and purpose.

To serve as the basis for assessing translation quality, we used the following specification-based definition developed within our research group:

> A quality translation achieves sufficient accuracy and fluency for the audience and purpose, while, in addition, meeting all other negotiated specifications that are appropriate to meet end-user needs.

This novel definition, which goes beyond a strictly industry-neutral, ISO 9000 approach to quality, makes the implicit claim that translation quality cannot be assessed without pre-determined specifications about the *process* of translation and the resultant *product*. Building on this definition, even a target text that is somewhat awkward yet usable could be a quality translation, if it fully meets the agreed-upon specifications. To this end, we developed and tested a methodology for formalizing structured translation specifications to support post-editing assessment. This methodology involves two software applications which we have developed: the Ruqual Specifications Writer, which aids in the authoring of post-editing project specifications, and the Ruqual Rubric Viewer which provides a graphical user interface for filling out a machine readable rubric file. Since this project uses a rubric to assess quality, the name of the software is Ruqual, which is a blend of “rubric” and “quality.”

2 Previous Work

Up until the last ten years, very little research had been done on the subject of post-editing (Allen, 2003). However, advances in MT have prompted an increase of interest in the subject (Alves, 2003; Guerberof, 2009; O'Brien, 2002, 2005, 2011; Ramos, 2010; Rios et al., 2011; Specia et al., 2009, 2011).

Most previous studies have focused on post-editing effort and the quality of the raw MT target text. This post-editing effort may be defined in a number of ways; most notably the work of Krings (2001) divides post-editing effort into three categories: temporal, technical, and cognitive. Temporal effort measures how long it takes the post-editor to finish editing the target text, whereas technical effort measures the changes made to the MT-generated text. The cognitive load is difficult to measure because techniques designed to measure the thought processes of translators/post-editors often make the task of translating more difficult (O'Brien, 2005). However, O’Brien has found that a measure of cognitive effort can be obtained from other measures, such as comparing the differences between the changes of multiple post-editors and accepting pauses in the timed record of changes as an indication of increased cognitive activity (2005). Specia and Farzindar (2010) have also developed a system of measuring expected post-editing effort so that companies can estimate whether a particular machine translated text is worth sending to post-editors.

Measuring the effort—or the time it takes—to post-edit a text assumes that the post-edited target text has sufficient and similar quality in all cases compared and that the post-editor followed all of the procedures necessary for the project. Measuring strictly “the time it takes to post-edit a text” must be based on a definition of translation quality and a method of measuring it. The "transcendent" view of quality assumes that every translation exhibits the same high levels of accuracy and fluency. Once it is recognized that translation quality is not transcendent but relative, measuring post-editing effort is only useful when the specifications are the same. One machine translated text may be useless for a particular set of specifications while being suitable for another set of specifications. The amount of effort necessary to successfully post-edit a text in accordance with a set of specifications will probably change when the specifications change, even if the source, raw MT text, and post-editor are the same. Hence any measure of post-editing effort must be based on a foundation of defining and measuring quality applicable to raw and post-edited translation.

The approach to measuring the quality of post-editing espoused in this project rejects the transcendent view of quality. This project provides a way to organize the information necessary to clarify which quality factors are relevant to a particular post-editing project. One study may investigate how much time it takes to post-edit raw MT output into documentation strictly for internal use in a software company. Another study may involve producing translations for general public consumption. If the specifications are not explicitly stated, then the results of one study may be misinterpreted to be directly relevant to the subsequent project.

Moreover, if the specifications are stated but not organized, a comparison of two studies would be difficult. If one study concluded that post-editing should take less than 10 minutes to be cost-effective, then such a measure might discriminate against good post-editors who take 20 minutes to post-edit a text in a different study. The reason for the difference in time may have less to do with individual post-editors than it does with the project specifications. It takes more effort to post-edit a text for a general audience than it does for a small audience that has more background knowledge and is more tolerant of errors. Explicit, structured project specifications and quality measures based on them are needed to complement on-going research in post-editing effort.

Translation specifications and quality measures must not only be explicit, they must be reliable. If evaluators cannot agree on the quality of a translation, human, raw machine, or post-edited, then the notion of quality is useless.

Colina has proposed a rubric for assessing the quality of human translation in a healthcare environment. Colina's approach is compatible with the definition of quality used in this project (Colina, 2008). The TAUS Labs have recently developed a Dynamic Quality Evaluation Framework (TAUS Labs, 2012) that may be compatible with the approach in this paper, but it is not available to the public. The EU-funded QT Launchpad project (2012) is also working on translation quality assessment. Collaboration among these related efforts would be beneficial to the translation industry.

3 Structured Specifications

This project proposes a format for formalizing structured translation specifications in order to support post-editing assessment. The basic components of the formalized structured translation specifications (FSTS) format are derived directly from the recently published ISO document ISO/TS 11669 (General Guidance -- Translation Projects) and the status descriptors in the Linport STS format (Linport, 2012; Melby et al., 2011), based on the earlier Container Project.

As shown in Table 1, the top-level categories in the FSTS format are Linguistic (divided into Source Content Information and Target Content Requirements), Production tasks to be performed during the project, Environment requirements, and Relationships between the requester (sometimes called the client, although "client" is ambiguous) and the translation service provider.

A. Linguistic [1–13]

Source content information [1–5]

- [1] textual characteristics
    - a) source language
    - b) text type
    - c) audience
    - d) purpose
- [2] specialized language
    - a) subject field
    - b) terminology [in source]
- [3] volume (e.g. word count)
- [4] complexity (obstacles)
- [5] origin [of the source content]

Target content requirements [6–13]

- [6] target language information
    - a) target language
    - b) target terminology
- [7] audience
- [8] purpose
- [9] content correspondence
- [10] register
- [11] file format
- [12] style
    - a) style guide
    - b) style relevance
- [13] layout

B. Production tasks [14–15]

- [14] typical production tasks
    - a) preparation
    - b) initial translation
    - c) in-process quality assurance
- [15] additional tasks

C. Environment [16–18]

- [16] technology
- [17] reference materials
- [18] workplace requirements

D. Relationships [19–21]

- [19] permissions
    - a) copyright
    - b) recognition
    - c) restrictions
- [20] submissions
    - a) qualifications
    - b) deliverables
    - c) delivery
    - d) deadline
- [21] expectations
    - a) compensation
    - b) communication

Table 1. List of 21 formalized structured translation specifications (FSTS).

The Source category describes the source content. The Target category is concerned with the language into which the material is translated and various other requirements for how the translation is to be carried out. The Production category lists the tasks to be performed during the translation project. The Environment category includes any technology that must be used, all reference materials that must be consulted by either software or human, and any security requirements, such as the need to conduct the work in a particular location. The Relationships category refers to the project expectations and work requirements for all team members, including the post-editor.

The five FSTS categories (Source, Target, Production, Environment, and Relationships) arrange the 21 translation parameters into logical groups, as is shown in Table 1. All parameters have two attributes that assist in determining its importance for a particular project: Status and Priority. The value of the status attribute can be one of four options: Incomplete, Not Specified, Proposed, and Approved.

One of the key components of the development of our methodology was the use of "Directives," or prose descriptions of specific instructions that could be assessed by an evaluator during the translation workflow process. Our methodology makes a distinction between *process-oriented* directives, or instructions to the post-editor concerning the steps he or she should follow while modifying the translation, and *product-oriented* directives, which relate to the final state of the target text.

The Ruqual Specifications Writer allows for the development of post-editing project specifications which are both process- and product-oriented. In its design, several parameters and attributes in the FSTS take a list of directives as their value. A directive has two attributes: Request and Priority. The request consists of natural language content describing the post-editor's task. The priority indicates how important it is that the request be fulfilled. Each directive can be modified based on project specifications.

The FSTS naturally support the generation of a rubric for evaluating post-editing that can handle a high degree of variability in the specifications of various projects. The rubric developed in our methodology, the Ruqual Rubric Viewer, is composed of a list of directives pertaining to the top-level FSTS categories previously mentioned. When using the rubric for assessment, an evaluator simply specifies whether a particular directive was fulfilled or not. If it was fulfilled, the value of the priority is awarded; otherwise, no points are awarded. The final score for a given category is the number of points received divided by the number of points possible.

With these tools it is possible to write translation project specifications and consequent rubrics that will allow non-experts to quickly and straight-forwardly assess the translation quality of post-edited texts.

The software developed for this research is hosted as an open source Google Code project at: http://code.google.com/p/ruqual/. Collaboration with other projects and extensions of the software are welcome.

4 Study Design

In the structured assessment of our methodology and accompanying Ruqual software, a Japanese source text was translated by Google Translate (Google, 2012) to produce a raw machine translated text. With attention to real post-editing data, five different potential post-edited texts were developed from the machine translation to simulate the work of five fictional post-editors, whom we named Editors A-E. Five different scenarios describing the translation process experiences of the five fictional post-editors were also developed. Errors were purposefully introduced into the post-editors' scenarios such that some violated process-oriented directives while others violated product-oriented directives. The source text, raw machine translation, and five post-edited texts are shown in the appendix.

Space limitations for this paper do not allow inclusion of the full FSTS used in this experiment. Some of the key elements of the specifications were that the translation was for a general audience, that it should be fluent and not obviously a translation, that a particular bilingual glossary must be used, and that the translation product must be delivered by a certain date and in a particular format. The authors are quite aware that in many post-editing environments the translation can be less than fluent and can be an obvious translation.

The definition of a non-expert assessor in this study was an individual who was:

1. A non-native speaker of the source language, but who had studied the source language for at least two years;
2. A native speaker of the target language;
3. A high school graduate or higher; and
4. A novice in the professional translation industry.

The focus of the study was reliability, that is, how consistent the non-expert assessors were with each other. In order to assure that the assessment was also reasonably valid, the assessments were compared with those of an expert assessor.

In total, 17 non-experts provided complete assessments of the five work products of the simulated post-editors A-E. The data were gathered via a questionnaire that was accessible from the Ruqual website (ruqual.gevterm.net).

The first portion of the questionnaire asked for some basic demographic information, and then participants were directed to an instructions page that included four items:

1. A video demonstrating the Ruqual Rubric Viewer;
2. A text walk-through with the same content as the video in case the participant lacked the software necessary to display the video;
3. A location from which to download the source materials and terminology files; and
4. A link to a zipped version of the Ruqual Rubric Viewer.

Participants were instructed to familiarize themselves with the software and source materials before proceeding with the questionnaire.

The second portion of the questionnaire presented the work of each of the five fictional post-editors in random order. The evaluators were instructed to assess the performance of each post-editor independently.

5 Analysis

The analysis of the data was twofold. First, we examined reliability among the non-expert evaluators as well as the concordance of the non-expert evaluators with an expert evaluator. Second, we examined the similarities and differences among the five fictional post-edited target texts and a human translation provided by an expert human translator.

Figure 1 shows a comparative box plot of scores given on a scale ranging from 0.0 to 1.0 for the five post-editors ordered by median score.

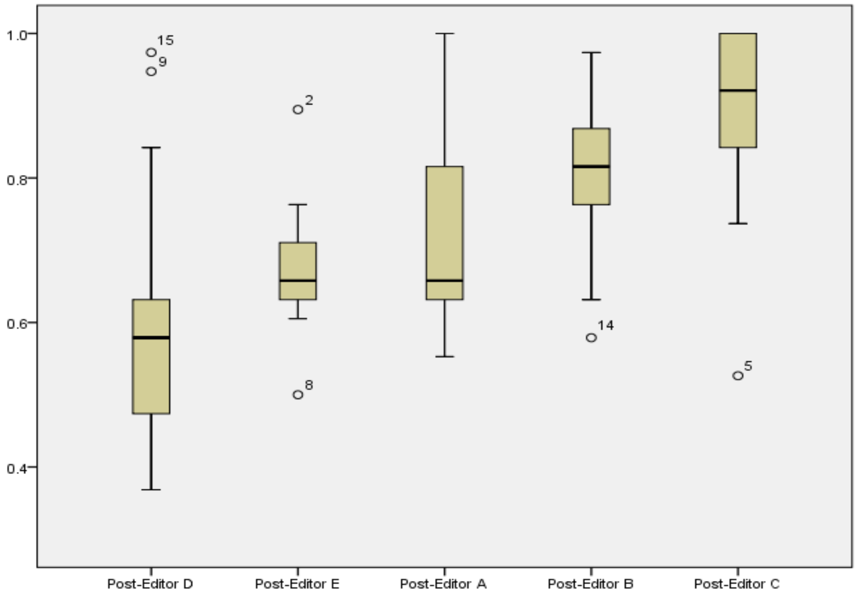


Figure 1. Scores Given by the Non-Expert Evaluators for Each Post-Editor Ordered by Median Score.

Overall, non-expert evaluators gave the highest scores to post-editor C and the lowest score to post-editor D, which is as would be expected based on the number of specifications these post-editors were designed to violate. Assessments for post-editors E spanned a much smaller range around the median than the scores for post-editors A, B, C, and D. It should be noted that no post-editor received a score lower than 0.35 from any grader, which could be due to the fact that all of the target texts were, in our opinion, reasonably grammatically correct.

In order to test reliability, which was the focus of the research question, we calculated the two-way random Intraclass Correlation Coefficients (ICC) as described by Shrout and Fleiss (1979).

ICC values can range from 0.0 to 1.0 analogous to percentages. The ICC is a measurement of the agreement between evaluators, or in other words, the percentage of variability in the scores that represents the quality of the post-editing. Using ICC values provides a measure of the agreement and consistency of the evaluations. The question of reliability in this case is not simply whether evaluators assigned the same relative scores to the post-editors, but to what degree they assigned the same scores. Since all 17 non-expert evaluators assessed all five fictional post-editors and these evaluators can be considered to be a sample of potential non-expert translation evaluators, the ICC values calculated utilized a two-way random effects model with evaluator effects and measurement effects. Table 2 shows the single and average ICC scores for the non-experts evaluators subdivided by rubric categories.

Target	0.167	0.773
Production	0.148	0.747
Environment	0.529	0.95
Relationships	0.607	0.963
Total	**0.426**	**0.927**

Table 2. Single and Average Intraclass Correlation Coefficients (ICC) for Non-Expert Evaluators.

The single ICC is a measure of the reliability of a single evaluator from this set of evaluators, if we were to accept his or her score alone. The average ICC indicates at the percentage of agreement among the evaluators with each other as a group.

The key statistic in Table 2 is the average ICC for the total score, which is ICC (2, 17) = 0.927. This is a strong indicator that the non-expert evaluators were reliable as a group. However, the single ICC for the same category was only ICC(2, 1) = 0.426 suggesting that if one evaluator was to be selected from this set, he or she would be expected to be reliable only about 43% of the time.

Looking at the rubric categories, there appears to be a split between the Target/Production specifications and Environment/Relationships specifications. This is worth noting because the specifications as constructed for this research generally include product-oriented directives in Target/Production and process-oriented directives in Environment/Relationships. The evaluators might have had an easier time agreeing on whether a post-editor followed the specified processes than they did deciding whether a particular text sufficiently corresponded with another text.

In addition to reliability there is also the question of whether non-expert evaluators were assessing the post-editors in a manner similar to how an expert would do so. The expert evaluator's assessments are provided in Table 3 along with 95% confidence intervals for the non-experts.

	A	**B**	**C**	**D**	**E**
Expert Scores	0.447	0.684	0.763	0.315	0.605
Non-Expert Upper Confidence Limit	0.784	0.863	0.957	0.693	0.716
Non-Expert Lower Confidence Limit	0.648	0.755	0.829	0.514	0.627

Table 3. Expert Evaluator's Scores and 95% Confidence Intervals for the Non-Expert Evaluators' Scores.

The expert evaluator assessed post-editor C as the best and D as the worst, with post-editor B assessed as next best after post-editor D. The non-expert evaluators matched these rankings. Although the expert and non-experts reversed the rank order of post-editors A and E, it should be noted that there was no statistical difference between post-editors A and E for the non-experts. These data suggest that the goal of providing the evaluators with simulated post-editors showing distinct differences and a progression from higher quality to lower quality was achieved.

In addition, we calculated a coefficient of concordance for each non-expert evaluator and the expert evaluator. Two evaluators showed a strong concordance with the expert and ten evaluators showed a moderate concordance. One evaluator showed a weak concordance and four evaluators showed little or no concordance. The four evaluators who showed no concordance with the expert, were also the evaluators who gave the most extreme evaluation scores (evaluators 2, 5, 9, and 15 in Figure 1). This suggests that these evaluators were perhaps less skilled or less trained (i.e., they did not go through the prescribed training) than the other evaluators.

However, none of the expert evaluator's scores fell within the confidence intervals of the non-experts' scores. In fact, the expert provided a lower score than the non-experts in all cases, indicating that the expert may have allowed the post-editors less leeway in evaluating their work products. Since the expert would have had a better understanding of the importance of following proper procedures and fully meeting the translation specifications, perhaps the expert was either more inclined to find fault with the performance of the fictional post-editors or was more aware of the failures present in the text and scenario.

As an addition to our study, a second expert was sought out to provide a human translation of the source text used in the study without reference to the raw machine translated text. In fact, the second expert was only given the source text and specifications. The purpose of requesting an expert human translation was to obtain a reference translation for the source text. Since none of the post-edited texts were intended to meet all of the specifications, it was worthwhile to identify how closely these fictional post-edited texts were to an actual human translation. If the human translation did in fact meet the specifications, and if post-editing is worthwhile, then the post-edited text should have been generally similar to the HT reference text.

In our comparison it appeared that the human translator took advantage of the flexibility provided by the specifications that allowed for some awkward sentences as long as the target text fulfilled its purpose for the intended audience. (The expert human translation is also shown in the appendix.) The human translator also rendered some sentences in a way that typically would be described as run-on sentences, but these sentences closely matched the flow of the source text. In fact, such sentences may facilitate automatic alignment and processing better than the sentence breaks provided by the machine translation. Overall, it appeared that post-editor C and the human translator were generally similar, but the requirement to not change sufficiently translated phrases in the initial machine translation could have limited the latitude of a post-editor.

6 Results

Overall, the research results support the hypothesis that non-expert evaluators can reliably assess the quality of fictional post-edited translations when taken as a group. This is a promising outcome since it shows that it is possible to obtain agreement about the quality of post-editing when using formalized structured translation specifications and multiple evaluators. Moreover, the fact that a majority (12 out of 17) of the non-expert evaluators showed at least moderate concordance with an expert evaluator suggests that there was evidence of the validity of the non-experts' evaluations.

Consequently, we assert that using FSTS specifications provides both a practical and realistic basis for evaluating the quality of post-edited texts. Therefore, if appropriate specifications are provided and structured via the Ruqual tools developed in this research, then evaluators can be expected to reach generally reliable and valid conclusions.

Finally, the fact that the text judged to be the work of the best post-editor was similar to the text produced by a human expert translator supports the assertion that post-edited MT text can be of sufficient quality to compete with HT alone.

7 Conclusions and future work

Overall, the research results provide evidence that non-experts can reliably assess the quality of post-edited machine translation relative to structured specifications. Further studies need to be conducted using the same approach to determine the effect of more training for the assessors and the effect of more specific rubrics.

More important than the particular experiment described in this paper are the methodology and tools used in the experiment. We anticipate working with other teams to conduct a series of experiments using various source texts, alternative machine translation systems, and widely varying project specifications, but applying the same methodology as in this study, and including the same standard set of translation parameters from

ISO/TS 11669 as well as the same definition of translation quality for human and machine translation. This will allow meaningful comparison of results. If there are problems with the translation parameters, suggestions should be made to the ISO project 11669 team as they prepare the next version. Extensions to ISO 11669, such as much more detailed and narrow assessment specifications, can be developed. Other work in post-editing, such as measures of effort, needs a widely used, reliable approach to translation quality assessment.

References

Allen, Jeffrey. 2003. Post-editing. In H. Somers (Ed.), *Computers and translation: A translator's guide* (297-317). Philadelphia: John Benjamins.

Alves, F. 2003. *Triangulating Translation*. Amsterdam: John Benjamins.

Colina, Sonia. 2008. Translation Quality Evaluation: Empirical Evidence for a Functionalist Approach. *The Translator*, 14: 97-134.

Google. 2012. Google Translate. http://translate.google.com/. (Accessed 2012).

Guerberof, Ana. 2009. Productivity and quality in MT post-editing. Paper presented at the MT Summit XII Workshop: *Beyond Translation Memories: New Tools for Translators*. Ottawa, Ontario, Canada.

Hutchins, John. 2007. Machine Translation: A concise history. In C. Wai (Ed.), *Computer aided translation: Theory and practice*. Hong Kong: Chinese University of Hong Kong.

ISO/TS 11669: Translation projects. General guidance. 2012. ISO. Geneva, Switzerland.

Krings, Hans P. (Ed.) 2001. *Repairing texts: Empirical investigations of machine translation post-editing processes*. Kent, OH: Kent State University Press.

Linport. 2012. Linport: The Language Interoperability Portfolio Project. www.linport.org. (Accessed 2012).

Melby, Alan K., Arle Lommel, Nathan Rasmussen & Jason Housley. 2011. The Container Project. Paper presented at the First International Conference on Terminology, Languages, and Content Resources. Seoul, South Korea.

O'Brien, Sharon. 2002. Teaching Post-Editing: A Proposal for Course Content. Paper presented at the 6th EAMT Workshop: *Teaching Machine Translation*. Manchester.

—. 2005. Methodologies for Measuring the Correlations between Post-Editing Effort and Machine Translatability. *Machine Translation,* 19: 37-58.

—. 2011. Towards predicting post-editing productivity. *Machine Translation,* 25:197-215.

QT Launchpad. 2012. Motivation webpage. http://www.qt21.eu/launchpad/. (Accessed 2012).

Ramos, Luciana. 2010. Post-Editing Free Machine Translation. Proceedings from the Ninth Conference of the Association for Machine Translation in the Americas (AMTA): *From a Language Vendor's Perspective*. Colorado.

Rios, Miguel, Wilker Aziz & Lucia Specia. 2011. TINE: A Metric to Assess MT Adequacy. Paper presented at the 6th Workshop on Statistical Machine Translation. Edinburgh.

Shrout, Patrick E. & Joseph L. Fleiss. 1979. Intraclass Correlations: Uses in Assessing Rater Reliability. *Psychological Bulletin,* 86: 420-28.

Specia, Lucia & Lucas Nuna Vieira. 2011. A Review of Machine Translation Tools from a Post-Editing Perspective. Paper presented at the 3rd Joint EM+/CNGL Workshop: *Bringing MT to the USER: Research Meets Translators*. Luxembourg.

Specia, Lucia & Atefeh Farzindar. 2010. Estimating Machine Translation Post-Editing Effort with HTER. Paper presented at the AMTA 2010 Workshop: *Bringing MT to the USER: MT Research and the Translation Industry*. Denver, Colorado.

Specia, Lucia, Dhwaj Raj & Marco Turchi. 2009. Machine translation evaluation versus quality estimation. *Machine Translation,* 24: 39-50.

TAUS Labs. 2012. About DQF webpage. http://www.tauslabs.com/dynamic-quality/about-dqf. (Accessed 2012).

Appendix

Source Text (from a 2009 source accessed June 2012 http://www.asahi.com/topics/%E3%82%A2%E3%83%83%E3%83%97%E3%83%AB.php):

解説

アップルとは

パソコンから携帯電話まで

マッキントッシュ（Ｍａｃｉｎｔｏｓｈ）などのパソコンや携帯音楽プレーヤー・アイポッド（ｉＰｏｄ）、携帯電話アイフォーン（ｉＰｈｏｎｅ）などを販売する、米コンピューターメーカー。世界中で直営店（アップルストア）やネット店を展開し、アイチューンズ（ｉＴｕｎｅｓ）ストアでは音楽や映画などの販売、アップストア（ＡｐｐＳｔｏｒｅ）ではｉＰｈｏｎｅ向けのソフト販売も行う。世界市場ではマイクロソフトのウィンドウズで動くパソコンが9割以上を占め、同社のパソコンは数%のシェアしか得ていないが、斬新なデザインや使い勝手の良さなどでファンを獲得し独自路線を歩んできた。カリスマ的な経営者スティーブ・ジョブズ氏の言動が常に世界中で注目されることでも有名。しかし彼は２０１１年10月に死去した。

Raw Machine Translated Target Text

Commentary
Apple and
From PC to mobile phone
Macintosh personal computer or portable music player such as iPod (Macintosh) (iPod), mobile phones to sell, such as iPhone (iPhone), the U.S. computer maker. Expand the net and shop (Apple Store), at (iTunes) store selling music and movies, also performs in software sales for the iPhone (App Store) Apple retail store in the world up. In the world market a personal computer running on Windows Microsoft accounted for more than 90%, PC company is not only get share of a few percent, has come a maverick won the fan, such as the difference between ease of use and innovative design. Well known for their words and deeds of Mr. Steve Jobs, the charismatic owner is always attention throughout the world. He died in October 2011, however.

Post-Edited Texts and Scenarios

Post-Editor A

Scenario:
When the post-editor received the source materials, he/she sent them to an acquaintance who converted them to MS Word 2003 format because the post-editor did not own MS Word 2007 or greater. After finished post-editing, the post-editor sent the post-edited target text to the acquaintance to have it converted to .docx format, but this delayed the project meaning that the finished product was returned on March, 28 2012.

Target Text:
Company Description
Apple
From PC to mobile phone
Apple is the American computer maker that markets the Macintosh computer series, the iPod MP3 player, and the iPhone smart phone. Around the world, Apple has opened an internet store and company stores (Apple Stores). Music and movies are sold at the iTunes store whereas at the App Store software for the iPhone is sold. In the world market personal computers running on Microsoft Windows account for more than 90%. However, Apple only controls a few percentage points of market share, but Apple has taken its own approach to consumer electronics and won fans for its ease of use and innovative design. The company is well known for the words and deeds of Mr. Steve Jobs, the charismatic CEO, who was always earning the attention of the world. However, he passed away in October of 2011.

Post-Editor B

Scenario:
After getting the source materials, the post-editor verified via email that he/she had received all of the source materials and would follow all of the specifications. The post-editor owns and uses MS Word 2007 and Acrobat Reader. The post-editor completed and returned the post-edited text on March, 24 2012. He/she also included the source text and machine translation. Finally, the post-editor deleted all of the project files after hearing back that that project was complete.

Target Text:
Commentary
What's up with Apple?
From computers to mobile devices
Apple Inc. is a consumer electronics manufacturer that sells personal computers (Macs), iPod portable music players, and mobile phones such as the iPhone at pure bead prices. They also have retail stores (Apple Stores) and an internet store. In the iTunes store they charge extra for music and movies, and at the App Store they sell sub standard software for the iPhone. In the world market, machines running on Windows make up more than 90% of all machines. Apple Inc., on the other hand, only controls a small percentage of the market. Apple Inc likes to do its own thing and has won some fans for its perceived ease of use and innovative design. Ultimately, the company gets a lot of attention for the words and deeds of Mr. Steve Jobs, the ostentatious owner, but he died in October of 2011.

Post-Editor C

Scenario:
Upon receipt of the source materials, the post-editor confirmed that he/she had received all of the source materials and would follow all of the specifications. The post-editor used Acrobat Reader to view the appleTerms.pdf file and MS Word 2010 to edit the target text. When finished post-editing, the post-editor returned the post-edited text alongside the source and machine translation on March, 23 2012. After completing the project, the post-editor deleted all of the related texts.

Target Text:
Company Overview
Apple
From PC to mobile phone
Apple is a U.S. computer maker that sells products such as the Macintosh personal computer, iPod portable music player, and mobile phones such as the iPhone. Apple has opened an internet store and company stores

(Apple Stores) across the world. At the iTunes store they sell music and movies, while at the App Store they sell programs for the iPhone. In the world market personal computers running on Microsoft Windows account for more than 90% of the market, whereas Apple has only a small percentage of market share, but Apple has marched to the beat of its own drummer and won fans via ease of use and innovative design. The company is well known for the words and deeds of Mr. Steve Jobs, the charismatic owner, who is always getting attention throughout the world. However, he died in October of 2011.

Post-Editor D

Scenario:
The post-editor never agreed to the specifications. Instead, the post-editor returned the post-edited target text (by itself) on March, 26 2012 before a formal agreement was complete. After the project was complete, it was discovered that the post-editor had posted a copy of the source materials and his/her translation on his/her blog for the public to write comments about.

Target Text:
Commentary
And Apple?
From personal computers all the way to mobile devices: Apple Inc. is in the business of consumer electronics. They make Macs, iPod portable music players, and the iPhone intelligent cell phone. They expanded their net shop and retail stores (Apple Stores); in the iTunes store they offer tunes and flicks, and at the App Store they have a place for applications that run on the iPhone. In the world market a personal computer generally is running on Microsoft Windows, which accounts for more than 90%. Apple Inc. is a maverick to its fans, who love its innovative design and dang good usability. The company is well known for its attention getting owner Mr. Steve Jobs, but he died in October of 2011.

Post-Editor E

Scenario:
Because the post-editor did not own MS Word (and did not realize that he/she could download Acrobat Reader for free), he/she sent the source materials to a friend asking for help. The friend was late returning the post-edited translation and did not include the original source text and raw machine translation. The post-editor hurriedly returned the finished post-edited translation on March 26, 2012, but he/she forgot to delete any of the project documents after the project was finished.

Target Text:
Company Overview
Apple
From PC to mobile phone
Apple is the U.S. computer maker that sells the Macintosh personal computer, iPod portable music player, and mobile phones such as the iPhone. They have company stores (Apple Stores) the world over and an internet store; at their iTunes store they sell music and movies, and at the App Store they sell software for the iPhone. In the world market, computers running Microsoft Windows account for more than 90%, while Apple has only a small percentage of the market, but Apple has followed its own path and won fans for its ease of use and innovative design. The company is well known for the words and deeds of Mr. Steve Jobs, the charismatic CEO, who is always getting attention throughout the world. He died in October of 2011, however.

Expert Human Translated Reference Text:

Commentary
Apple
From Personal Computers to Mobile Phones
Apple is a US computer manufacturer that sells Macintosh and other personal computers, the portable music player iPod, the mobile phone iPhone, and other products. Apple operates Apple Store outlets worldwide as well as an Internet shop, and sells music , movies, and other media at the iTunes store, and software for the iPhone in the Apps Store. Personal computers that operate Microsoft Windows account for over 90% of the world market, and Apple computers have only a small percentage of the market share, but due to their novel designs, ease of use, and other features, Apple computers have acquired fans and the company has walked an independent path. The words and actions of the charismatic CEO Steve Jobs were famous, being heard and seen worldwide. However, Steve Jobs passed away in October, 2011.

Learning to Automatically Post-Edit Dropped Words in MT

Jacob Mundt, Kristen Parton, Kathleen McKeown
Department of Computer Science
Columbia University
New York, NY 10027
jmm2328@columbia.edu,
{kathy,kristen}@cs.columbia.edu

Abstract

Automatic post-editors (APEs) can improve adequacy of MT output by detecting and reinserting dropped content words, but the location where these words are inserted is critical. In this paper, we describe a probabilistic approach for learning reinsertion rules for specific languages and MT systems, as well as a method for synthesizing training data from reference translations. We test the insertion logic on MT systems for Chinese to English and Arabic to English. Our adaptive APE is able to insert within 3 words of the best location 73% of the time (32% in the exact location) in Arabic-English MT output, and 67% of the time in Chinese-English output (30% in the exact location), and delivers improved performance on automated adequacy metrics over a previous rule-based approach to insertion. We consider how particular aspects of the insertion problem make it particularly amenable to machine learning solutions.

1 Introduction

Automatic post editors (APEs) use an algorithm to correct or improve the output of machine translation (MT). While human post editors have the intrinsic advantage of human linguistic knowledge, automatic post editors must have some other advantage over the MT system to be able to make improvements. The APE may have access to additional resources, either in the form of deeper contextual information or analysis unavailable to the decoder. Knight and Chander (1994) used additional analysis performed on the completed MT sentence to select determiners, while Ma and McKeown (2009) used redundancy in a question-answering task to help select better translations for verbs than were available in the MT phrase table. The APE may also have more knowledge about the specific translation goals of the system, allowing it to make different translation choices to better address those goals, even when selecting from the same phrase table. While MT systems trained on Bleu (Papineni et al., 2002) aim for fluency, Parton et al. (2012) used automatic post editing to adapt a black box MT system to prefer adequacy over fluency in a cross lingual question answering (CLQA) task where adequacy judgments determined task success.

Our motivation for improving adequacy is also CLQA, in our case over web forum data, as part of a new DARPA (Defense Agency Research Projects Agency) sponsored program called BOLT. CLQA system performance is evaluated by human relevance judgments comparing retrieved, translated passages to predetermined nuggets of information. As in Parton et al. (2012), an inadequate translation can cause an otherwise relevant passage to be judged irrelevant, so adequacy of MT is crucial to task performance. A critical problem in task-embedded translations is deletion of content words by MT systems and this is the focus of our work. Specifically, we are concerned with content words that are either translated into function words, or not translated at all in the MT output. These types of deletion are common in MT systems as a tradeoff to balance fluency and adequacy; Parton et al. (2012) detected these types of errors in 24% to 69% of sentences, with higher numbers of errors for web text over newswire copy. In our test sets, we also saw higher error rates for Chinese sources over Arabic.

Reference:
France and Russia are represented at both levels at the meeting...
MT:
It is both France and Russia at the meeting...

Figure 1. The MT drops the words "both levels", but the rephrasing of the rest of the sentence, while still expressing that France and Russia are at the meeting, presents no good place to reinsert "both levels".

A major challenge in automatic post editing, once the correct translation of a deleted word is found, is locating an insertion location that *maximizes adequacy*. This is a difficult problem for two reasons: first, the missing word was often dropped specifically to preserve fluency (to maximize the language model score). Additionally, phrases adjacent to a dropped word will typically be chosen to maximize fluency without the dropped word, as in Figure 1.

Parton et al. (2012) compare a rule-based automatic post editor with a feedback automatic post editor and for the rule-based approach use a simple alignment-based heuristic, inserting dropped content words adjacent to a partial translation if available, or between the translations of the dropped words' neighbors. In cases where the neighbors are not aligned to adjacent locations in the MT output, the correction is discarded. These heuristics provide reasonable results when translating between languages with similar word orders for the word being inserted and surrounding words. However, they can perform poorly in other cases; in translations from Arabic to English, subjects are often inserted after their verbs when the Arabic word order is VSO.

As an alternative to this heuristic, we present an approach for learning insertion positions from grammatical and positional features of the source sentence and aligned MT output. Since no gold standard training data is available for this problem, we also present a novel approach to generate high-adequacy insertion locations using reference translations. This method allows for better insertions of deleted words in languages with differing word order, improving adequacy of edited sentences. Further, in cases where Parton et al's heuristic method fails to determine an insertion point, this method can still succeed, allowing APE corrections to be applied in 14% more cases than their approach. Our evaluation using Chinese-English and Arabic-English MT systems shows that our insertion system can improve automated and human adequacy metrics in certain cases, when compared with both the original MT output and heuristic insertion.

2 Related Work

Our work builds on Parton et al. (2012) who compellingly show that a *feedback* and *rule-based* APE each have different advantages. The feedback post editor adds several potential corrections to the MT phrase table and feeds the updates back into another pass through the MT decoder, while the rule-based editor inserts the top-ranked correction directly into the original MT output. They found while the feedback system was preferred by the TERp (Snover et al., 2009) and Meteor (Denkowski and Lavie, 2011) automated adequacy metrics, the rule-based system was perceived to improve adequacy more often by human reviewers, often at the expense of fluency, noting that "with extra effort, the meaning of these sentences can usually be inferred, especially when the rest of the sentence is fluent." Our work attempts to increase adequacy through better insertion.

Previous general APE systems target specific types of MT errors, like determiner selection (Knight and Chandler, 1994), grammatical errors (Doyon et al., 2008), and adequacy errors (Parton et al. 2012). In contrast, fully adaptive APE systems try to learn to correct all types of errors by example, and can be thought of as statistical MT systems that translate from bad text in the target language to good text in the target language (Simard et al., 2007; Ueffing et al., 2008; Kuhn et al., 2011).

Similarly, Dugast et al. (2007) present the idea of statistical post editing, that is, using bad MT output and good reference output as training data for post editing. As their system proves more adept at correcting certain types of errors than others, they suggest the possibility of a hybrid post editing system, "breaking down the 'statistical layer' into different components/tools each specialized in a narrow and accurate area," which is similar to the approach followed in this paper. Isabelle et al. (2007) also use learning methods to replace the need for a manually constructed post editing dictionary. While they study a corpus of MT output and manually post-edited text to derive a custom dictionary, our system attempts to learn the rules for a specific type of edit: missing word insertion.

Taking a statistical approach to system combination, Zwarts and Dras (2008) built a classifier to analyze the syntax of candidate translations and use abnormalities to weed out bad options. Our classifier could be seen as a special case of this, looking for an area of bad syntax where a word was potentially dropped. As noted though, the MT system's language model often "patches up" the syntax around the missing word, leading to areas that are syntactically valid, though inadequate.

The TER-Plus metric (Snover et al., 2009) provides a variety of techniques for aligning a hypothesis to a reference translation, as well as determining translation adequacy amongst deletions and substitutions. While we use TER-Plus as a metric, we also use it as a guide for determining where a missing word should be inserted to maximize adequacy against a reference. While our effort focuses on learning the highest adequacy insertion from examples with reference translations, there is significant work in trying to assess adequacy directly from source and target, without references (Specia et al., 2011; Mehdad et al., 2012).

3 Method

The APE has 3 major phases: error detection, correction, and insertion. The first two phases are performed identically as described in Parton et al. (2012) and will be summarized briefly here, while the third phase differs substantially and will be described in greater detail.

3.1 Input and Pre-processing

We constructed two separate pipelines for Arabic and Chinese. The Arabic data was tagged using MADA+TOKEN (Habash et al., 2009). Translated English output was recased with Moses, and POS and NER tags were applied using the Stanford POS tagger (Toutanova et al., 2003) and NER tagger (Finkel et al.,2005).

For Chinese data, POS tags were applied to both source and output using the Stanford POS tagger (Toutanova et al., 2003).

3.2 MT systems

The Arabic MT system is an implementation of HiFST (de Gispert et al., 2010) trained on corpora from the NIST MT08 Arabic Constrained Data track (5.9M parallel sentences, 150M words per language). The Chinese MT system is the SRInterp system, developed by SRI for the DARPA BOLT project, based on work discussed in Zheng et al. (2009). It was trained on 2.3 million parallel sentences, predominantly newswire with small amounts of forum, weblog, and broadcast news data.

3.3 Error Detection and Correction

Errors are detected by locating mistranslated named entities (for Arabic only) and content words that are translated as function words or not translated at all, by looking at alignments and POS tags (Parton and McKeown, 2010).

Arabic error corrections are looked up in a variety of dictionaries, including an MT phrase table with probabilities from a second Arabic MT system, Moses (Koehn et al., 2007), using data from the GALE program available from LDC (LDC2004T17, LDC2004E72, LDC2005E46, LDC2004T18, LDC2007T08, and LDC2004E13). Secondary sources include an English synonym dictionary from the CIA World Factbook[1], and dictionaries extracted from Wikipedia and the Buckwalter analyzer (Buckwalter, 2004). Arabic additionally uses a large parallel background corpus of 120,000 Arabic newswire and web documents and their machine translations from a separate, third Arabic MT system, IBM's Direct Translation Model 2 (Ittycheriah 2007).

Chinese corrections are looked up in the phrase table of our Chinese MT, SRI's SRInterp system (Zheng et al., 2009), and also in a dictionary extracted from forum data, Wikipedia and similar sources (Ji et al., 2009; Lin et al., 2011).

3.4 Synthesizing a Gold Standard

Once an error is detected and a high-probability replacement is found, it must be inserted into the existing MT output. The straightforward solution is to use standard machine learning techniques to adapt to the translation errors made by a specific MT system on a specific language, but doing this is

[1] http://www.cia.gov/library/publications/the-world-factbook

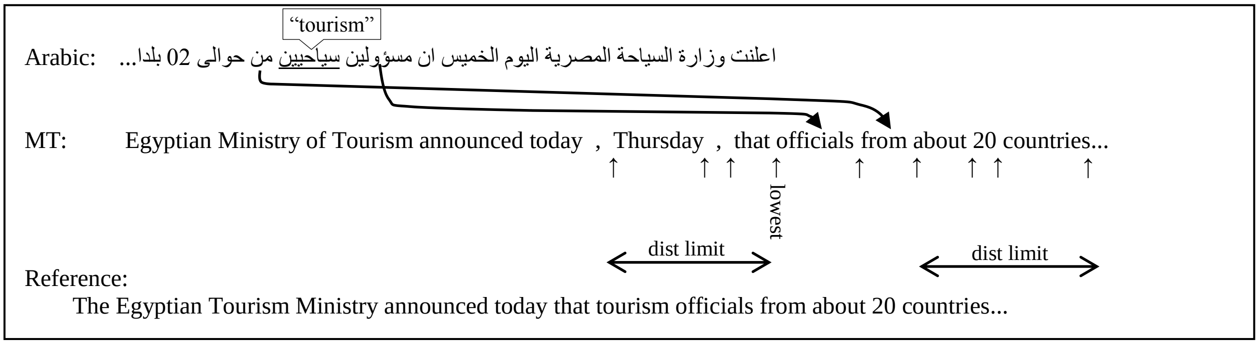

"tourism"
Arabic: اعلنت وزارة السياحة المصرية اليوم الخميس ان مسؤولين سياحيين من حوالى 02 بلدا...
MT: Egyptian Ministry of Tourism announced today , Thursday , that officials from about 20 countries...
lowest
dist limit
dist limit
Reference:
The Egyptian Tourism Ministry announced today that tourism officials from about 20 countries...

Figure 2. Synthesizing gold standard training data. The APE selects the nearly-correct alternative "tourist" for "سياحيين", and then TERp scores are evaluated at several potential insertion locations, up to a defined distortion limit from the source word's neighbors. The gold standard location is chosen as the one with the best (lowest) TERp score; here, the location is not between its neighbors, but before both of them.

complicated by the lack of training data. One option would have been to have human annotators select insertion locations for all the corrections detected above.

We took a different approach and elected to synthesize gold standard data. Since we had reference translations for our translation data (4 references for each Arabic sentence and 1-4 for each Chinese sentence), we exploited these sentences to find highly-probable correct insertion locations for each correction.

The TER-Plus metric (Snover et al., 2009) generates an adequacy score by penalizing deletions, insertions, substitutions, and shifts, in addition to allowing stem matches, synonym matches, and paraphrasing. This often allows it to calculate a set of shifts that largely align MT output to a reference, even when the MT output uses significantly different words and ordering.

If the missing word appears in any of the reference translations, TER-Plus is evaluated repeatedly, comparing that reference to the MT output with the missing word inserted at each possible insertion point, to find the location that is most aligned to the reference—the location with the lowest TER-Plus score (Figure 2). Similar to many statistical MT systems, we impose a hard distortion limit, trained on development data, that prevents words from moving more than a set amount from their neighbors' aligned output phrases. In fact, the insertion heuristic presented in Parton et al. (2012) can be thought of as having a distortion limit of 1.

Although the detected gold standard locations typically correspond with human judgments of "correctness" on where a missing word should be inserted, another view is that the classifier is learning to insert at the location that maximizes the TER-Plus score for the output sentence, which should at least raise the score over the heuristic method. It should be noted that not all sentences with detected errors will generate valid gold standard insertion locations. When the missing word does not appear in any of the reference translations, we discard the sentence from our insertion training data and do not attempt to find the highest TERp insertion.

3.5 Training and Insertion

Once we have a set of synthesized gold standard training data, a standard MT classifier can be trained to recognize good insertion locations. We used the BayesNet classifier from Weka (Hall et al., 2009). In addition to giving good results on recognition of individual insertions, it also reports classification probabilities rather than binary output. Since we have to choose amongst a number of insertion locations, this allows us to choose the highest confidence insertion location.

Machine learning is particularly well-suited to this problem. It allows easy adaptation to different languages and MT systems. Secondly, by tuning the system for high recall, we can bias the system towards making edits rather than leaving the sentence unchanged. In adequacy-focused tasks, leaving the sentence without a content word is often a poor choice, and an incorrect insertion location, if not too far from the correct point, can result either in improvement, or in no perceived change: as noted, humans are good at making sense of misordered translations. Of course, a bad insertion can degrade accuracy, but prediction errors occur more

	N	**error**	**edit RB**	**edit ML**	**gold**
Train					
Arabic	4115	54%	37%	-	842 (21%)
Chinese	6318	63%	28%	-	679 (11%)
Test					
Arabic	813	60%	41%	47%	168 (21%)
Chinese	1470	58%	25%	31%	201 (14%)

Table 1. Data details, showing the total number of sentences in each set (**N**), the percentage with a detected dropped or mistranslated word error (**error**), and the percent of sentences that were edited by the rule-based APE (**edit RB**) and the adaptive APE (**edit ML**). Note that only 10-20% of data can be used as synthetic gold standard data (**gold)** for machine learning. For test data, the adaptive post editor is able to edit more sentences than the heuristic rule based one.

	N	**exact**	**within 1**	**within 3**	**mean error**
Arabic	168	32%	52%	73%	1.81
Chinese	244	30%	46%	67%	2.32

Table 2. Classifier accuracy when determining word insertion location.

often on sentences that already have poor translations.

The features used for each potential insertion location are positional and syntactic:

- **Insertion point location**: relative and absolute location where insertion is being considered.
- **Neighbor offsets**: relative offset from the English phrases aligned to the word's source language neighbors.
- **Partial translation offset**: relative offset from a partial translation, for cases where a content word was translated as a function word.
- **Part-of-speech**: POS tag of the left and right neighbors of the insertion location, and bigrams of these neighbors and the POS of the word being inserted.
- **Simplified part-of-speech**: same as above, but POS tags are mapped to a simple, language agnostic set first.

Feature selection was performed on our original feature set using Weka's Chi Squared method, which indicated that the offset and POS unigram features were the most useful. We noticed that for our training set size, the POS bigram features led to overfitting and poor results on unseen data. By creating a smaller set of simplified tags and tag bigrams, we were able to retain some very shallow syntactic information while avoiding overfitting.

To use the trained classifier for insertion on test data, we simply run the classifier on each possible insertion point (within the hard distortion limit) for a missing word, and choose the insertion point with the highest positive confidence as the predicted insertion location.

4 Experiments

The Arabic training data consisted of 4115 sentences sampled from past years of the NIST Arabic-English task MT02 – MT05, each with 4 reference translations, and the Arabic test data was 813 sentences from the NIST MT08 newswire set. The Chinese training data consists of 6318 sentences, combined from forum, weblog data, and newswire data from NIST Chinese MT08 eval set and the DARPA GALE project. The Chinese test data is the NIST Chinese MT06 eval set, 1470 sentences. All data had at least one reference, and some sources included up to four.

We tested three automated metrics on the baseline MT output, output from the original rule-based APE described in Parton et al. (2012), and output from the APE with adaptive insertion on both Chinese and Arabic. Metrics are BLEU (Papineni et al., 2002), Meteor and TERp. Since BLEU is based on strict matching of bigrams, we do not expect post editing to improve the BLEU score in most cases, since it is rare that both the word inserted and its neighbors match the reference translation exactly. Meteor and TERp include adequacy and so should be more representative of the performance of our improved insertion algorithm. Note that TERp was also used to train our insertion system as well; one way of viewing the classifier is as a predictor for high-TERp insertion locations,

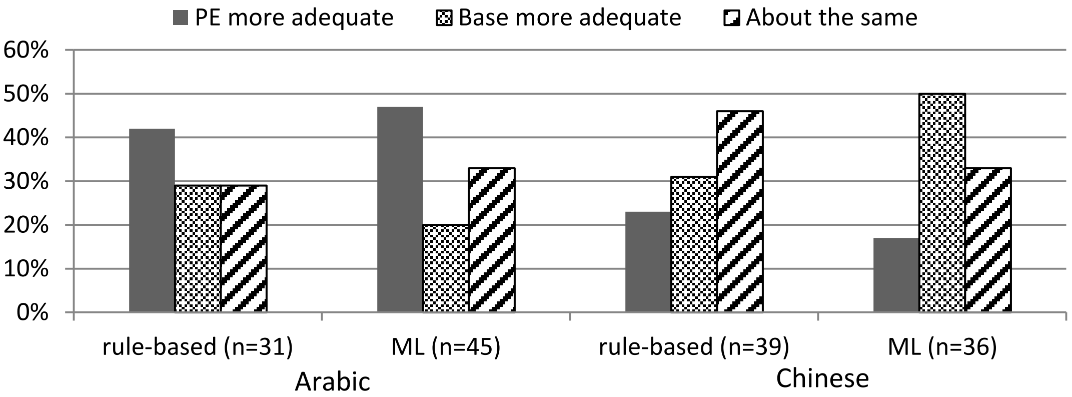


Figure 3. Human judgements on automatically post edited sentences. For each language, the results for the rule-based heuristic insertion algorithm is shown, along with our probabilistic ML approach. The total number of human comparisons performed is shown for each experiment.

trained on insertion locations that were shown to maximize TERp scores in the training set.

We also report the classifier results, showing what percentage of sentences were used to generate the synthetic gold standard, how often our classifier predicted the gold standard answer, and the average difference between our predicted insertion location and the gold standard.

5 Results

We are able to generate gold standard data for around 10-20% of the data using the TERp-based method described above, depending on the specific language (Table 1). The remaining cases do not have synthetic gold standard data, because it was not possible to align the word to be inserted with any of the provided references.

One clear advantage of the machine learning-based post editor is the ability to edit more sentences, as seen in Table 1. The rule-based editor cannot edit a sentence when the neighbors of the dropped word in the source are aligned to non-adjacent words in the MT output. The classifier in the adaptive editor always returns the highest-likelihood location within the distortion limit.

Turning to actual classifier accuracy, the exact gold standard insertion location is predicted 30% of the time in Chinese and 32% of the time in Arabic (Table 2). This is a meaningful result, since this is a multiclass prediction problem (where the number of possible places to insert is always at least twice the distortion limit). Also, the classification problem is continuous in some respects. Getting an insertion location *near* the correct one is better than getting one far away. We can predict the answer within 3 of the gold standard location 67-73% of the time. The mean error (in words) from the correct location is under 2 for Arabic and slightly higher for Chinese.

A simple human comparison was also performed, presenting the base MT output and the output of the APE, along with a reference translation, to 6 human annotators, who were asked to judge whether the APE was more adequate, the baseline was more adequate, or that the two translations had about the same adequacy. The number of human comparisons performed is noted in Figure 3 for each experiment. While we have a small number of survey results, the ML approach is preferred 47% of the time in Arabic, versus 42% for the rule-based APE. The ML APE also degrades only 20% of the Arabic sentences, whereas the rule-based system degrades 29%. This suggests that some of the degraded sentences were degraded because of a correct word inserted in an incorrect location.

Both APEs do significantly worse overall in Chinese, but the ML APE performs more poorly than the rule-based APE, both on number of sentences improved and number of sentences degraded. There may be attributes of the Chinese language that make reinsertion more difficult, but Chinese also had nearly 20% less training data than Arabic, and this may indicate that the performance of the ML APE suffered because of this.

6 Conclusions and future directions

We showed that a statistical approach to reinserting missing words is a feasible tactic, often able to predict locations near the correct location and sometimes even predicting the insertion location exactly. Though the insertion problem did not have human labeled gold standard data, we were able to generate it from reference translations. We also showed that the statistical approach can edit slightly more sentences than the original heuristic APE, leading to more adequacy improvements. Initial human judgments indicate that the statistical method increases adequacy in Arabic when compared with the rule-based approach, but is unable to improve adequacy in Chinese, possibly due to limited training data.

One area to be investigated is other methods for generating training data. Our TERp-based method requires that the inserted word (or a stem/synonym) be in the reference translation, but more flexible approaches may be possible using source and target POS tags or even full parses. Even better would be an approach that does not rely on reference translations, since this requirement limits the amount of training data we can generate. While earlier attempts have shown that purposely deleting words from correct English sentences provides poor training examples (since the "missing" areas are not adjusted by the language model to appear fluent), it may be possible to post-process the sentences after deletion, or even delete words from source sentences and then translate them.

Additionally, one continuing problem with this approach is the inability to apply more complicated modifications near the insertion point beyond simple insertion and replacement. Learning to apply more complicated changes (deleting nearby function words, fixing tense, determiners, and agreement) may be possible with sufficient training data and may help to improve fluency, rather than focusing almost exclusively on adequacy as we did here. This would be especially helpful in sentences with insertions located in contiguous areas of the sentence.

References

Tim Buckwalter. 2004. Buckwalter Arabic morphological analyzer version 2.0. Linguistic Data Consortium, University of Pennsylvania, 2004. LDC Catalog No.: *LDC2004L02*.

Adrià de Gispert, Gonzalo Iglesias, Graeme Blackwood, Eduardo R. Banga, and William Byrne. 2010. Hierarchical phrase-based translation with weighted finite-state transducers and shallow-n grammars. In *Computational Linguistics*, volume 36, pages 505–533.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*.

Jennifer Doyon, Christine Doran, C. Donald Means, and Domenique Parr. 2008. Automated machine translation improvement through post-editing techniques: analyst and translator experiments. In *Proceedings of the 8th AMTA*, pages 346-353.

Loïc Dugast, Jean Senellart, and Philipp Koehn. 2007. Statistical Post-Editing on SYSTRAN's Rule-Based Translation System. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 220-223.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL*, pages 363–370.

Nizar Habash, Owen Rambow, and Ryan Roth. 2009. MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In *Proc. of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, pages 242–245.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, Volume 11, Issue 1.

Pierre Isabelle, Cyril Goutte, and Michel Simard. 2007. Domain adaptation of MT systems through automatic post-editing. *MT Summit XI*.

Abraham Ittycheriah and Salim Roukos. 2007. Direct Translation Model 2. In *Proceedings of NAACL HLT 2007*, pages 57–64.

Heng Ji, Ralph Grishman, Dayne Freitag, Matthias Blume, John Wang, Shahram Khadivi, Richard Zens and Hermann Ney. 2009. Name Translation for Distillation. *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*.

Kevin Knight and Ishwar Chander. 1994. Automated postediting of documents. In *AAAI '94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 1)*, pages 779–784, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondˇrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In ACL '07: Interactive Poster and Demonstration Sessions, pages 177–180.

Roland Kuhn, Jean Senellart, Jeff Ma, Antti-Veikko Rosti, Rabih Zbib, Achraf Chalabi, Loïc Dugast, George Foster, John Makhoul, Spyros Matsoukas, Evgeny Matusov, Hazem Nader, Rami Safadi, Richard Schwartz, Jens Stephan, Nicola Ueffing, and Jin Yang. 2011. Serial System Combination for Integrating Rule-based and Statistical Machine Translation. In *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*, Joseph Olive, Caitlin Christianson, John McCary (Eds.), Springer, pages 361-374.

Wen-Pin Lin, Matthew Snover and Heng Ji. 2011. Unsupervised Language-Independent Name Translation Mining from Wikipedia Infoboxes. In *Proc. EMNLP2011 Workshop on Unsupervised Learning for NLP*.

Wei-Yun Ma and Kathleen McKeown. 2009. Where's the verb?: correcting machine translation during question answering. In *ACL-IJCNLP '09: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 333–336, Morristown, NJ, USA. Association for Computational Linguistics.

Yashar Mehdad, Matteo Negri, Marcello Federico. 2012. Match without a Referee: Evaluating MT Adequacy without Reference Translations. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 171–180.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Kristen Parton, Nizar Habash, Kathleen McKeown, Gonzalo Iglesias, Adria de Gispert. 2012. Can Automatic Post-Editing Make MT More Meaningful? In *Proceedings of the 16th EAMT Conference*.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical phrase-based post-editing. In *HLT-NAACL*, pages 508–515.

Matthew Snover, Nitin Madnani, Bonnie J. Dorr, and Richard Schwartz. 2009. Fluency, adequacy, or hter?: exploring different human judgments with a tunable mt metric. In *StatMT '09: Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 259–268, Morristown, NJ, USA. Association for Computational Linguistics.

Lucia Specia, Najeh Hajlaoui, Catalina Hallett and Wilker Aziz. Predicting Machine Translation Adequacy. 2011. *Machine Translation Summit XIII*, September, Xiamen, China.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL-HLT*, pages 173–180.

Nicola Ueffing, Jens Stephan, Evgeny Matusov, Loïc Dugast, George Foster, Roland Kuhn, Jean Senellart, and Jin Yang. 2008. Tighter Integration of Rule-based and Statistical MT in Serial System Combination. In *COLING 2008*, pages 913-920.

Jing Zheng, Necip Fazil Ayan, Wen Wang, and David Burkett. 2009. Using Syntax in Large-Scale Audio Document Translation. In *Proc. Interspeech 2009*, Brighton, England.

Simon Zwarts, Mark Dras. 2008. Choosing the Right Translation: A Syntactically Informed Classification Approach. In *COLING 2008*, pages 1153-1160.

SmartMATE: An Online End-To-End MT Post-Editing Framework

Sergio Penkale **Andy Way**

Applied Language Solutions
Delph, UK
firstname.lastname@appliedlanguage.com

Abstract

It is a well-known fact that the amount of content which is available to be translated and localized far outnumbers the current amount of translation resources. Automation in general and Machine Translation (MT) in particular are one of the key technologies which can help improve this situation. However, a tool that integrates all of the components needed for the localization process is still missing, and MT is still out of reach for most localisation professionals. In this paper we present an online translation environment which empowers users with MT by enabling engines to be created from their data, without a need for technical knowledge or special hardware requirements and at low cost. Documents in a variety of formats can then be post-edited after being processed with their Translation Memories, MT engines and glossaries. We give an overview of the tool and present a case study of a project for a large games company, showing the applicability of our tool.

1 Introduction

The amount of content that needs to be translated and localised is increasingly growing (DePalma and Kelly, 2009). With the current focus on user-generated content and an increasing commercial interest in emerging economies, the contents which are available for translation and the amount of languages into which this content is published are set to continue increasing. However, the high costs associated with translation and localisation mean that only a fraction of this content actually ends being translated, even more so given the current global economic difficulties.

It is hardly surprising then that, as evidenced by SDL's acquisition of Language Weaver, Language Service Providers (LSPs) are turning to automation in a bid to reduce translation costs at the same time as increasing the volume of translated content. However, while large LSPs are benefiting from the increased productivity associated with state-of-the-art Statistical Machine Translation (SMT), this technology remains out of reach for smaller organizations and individual translators. In particular, a tool that integrates all of the components required in a typical translation workflow (cf. Figure 1 for a sketch, and Section 3 for details on each of the steps in this workflow), and which allows users to easily exploit MT and postedit its output is crucial to enable mass adoption of MT.

In this paper we present one such tool. SmartMATE (Way et al., 2011) is a self-serve translation platform which supports File Filtering, Machine Translation, Terminology management, and which has an integrated Editor Suite. Crucially, SmartMATE enables both individuals and companies to train an MT engine using their own data, at the press of just a few buttons. By doing so, SmartMATE effectively removes the main barriers against exploiting MT technology. Expensive hardware requirements and technical knowledge are done away with, and so is computational linguistics expertise. In addition, SmartMATE supports unique capabilities such as concurrent translation and proofreading, terminology-aware MT, and integrated QA control inside the editor. We present all of SmartMATE's

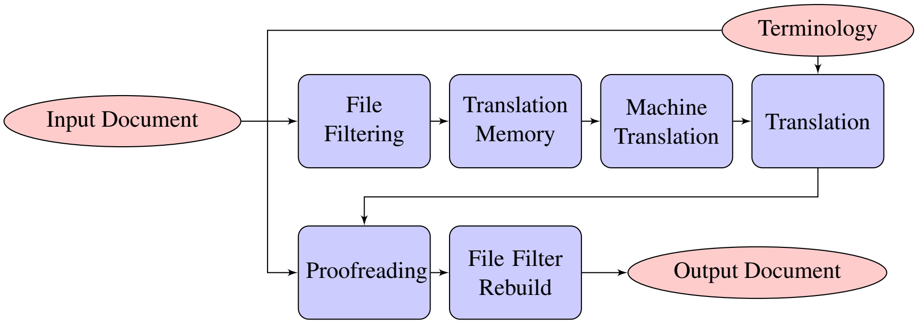


Figure 1: Typical translation workflow

capabilities, and discuss a case study of a large translation project carried out using our tool.

The remainder of this paper is organized as follows. Section 2 provides a brief review of translation platforms of a similar nature to the one presented in this paper. Section 3 presents SmartMATE and gives an overall introduction to all of its capabilities. In Section 4 we analyse a project currently being run for one of our customers using SmartMATE. We conclude and give avenues for future work in Section 5.

2 Related Tools

Although a few products which enable MT output to be postedited have been made available over the last few years, we are not aware of any tool which integrates all the capabilities offered by SmartMATE. Google Translator Toolkit[1] allows users to upload documents and pre-translate them using Google Translate. However, unlike SmartMATE only generic MT engines are used, providing no facility for the user to train an engine adapted specifically to their data. In addition, although terminology is supported in the post-editing environment, the MT engines are not aware of glossaries, making the pretranslated content unaware of the user's terminology requirements.

Unlike Google's offer, Microsoft Translator Hub[2] does enable user-specific engines to be created. It does not, however, provide postediting facilities, making the need for an external tool a requirement in order to allow a linguist to correct the MT output.

Finally, an interesting tool which finds itself in the opposite situation is PET (Aziz et al., 2012), which was designed specifically to post-edit the output of MT systems, and to collect various kinds of statistics from the process. However, the tool comprises only the editor part, and no actual MT services are provided.

3 SmartMATE

SmartMATE (Way et al., 2011) is an online self-serve translation platform. It is designed to be a one-stop portal where users can upload their Translation Memory (TM) files, and create user-customized MT engines trained using these TMs. It integrates all the capabilities needed in a typical translation workflow.

Figure 1 gives a sketch of a typical translation workflow in SmartMATE. Assume an input document which needs to be translated arrives. Since there is a variety of file formats in which this document can be encoded, it is first sent to File Filtering, which produces an XLIFF[3] (XML Localisation Interchange File Format) file containing only the translatable text, without additional elements such as images or page formatting information. Except for File Filtering, all of the components in SmartMATE take an XLIFF file as input and produce a modified one as output. This XLIFF can then optionally be sent through Translation Memory for leveraging of any previous translations, and through MT for segments which do not match any TM entry. At this stage, the document becomes available for editing. SmartMATE provides an online multi-user Editor Suite. Users can utilise the editor themselves to translate the document, or they might delegate this to a third party who receives an invitation email which enables

[1] http://translate.google.com/toolkit/

[2] http://hub.microsofttranslator.com/

[3] https://www.oasis-open.org/committees/xliff/

them to work on the document using the online editor. After translation has finished, the translated XLIFF file is sent back to File Filtering to recover the original file format. The following sections provide details on each of these components.

It is important to note that SmartMATE's terms and conditions explicitly state that any data uploaded into SmartMATE will be kept confidential. TMs, input documents, glossaries and MT engines are kept in the user's password-protected area, being unreachable by other users, and ALS will not exploit any of this data for other purposes without the users's consent.

3.1 File Filtering

SmartMATE accepts a wide range of input document formats, including Microsoft Office Suite file formats (e.g. .doc, .xls, .ppt), as well as other popular formats such as .rtf, .html, .ttx and .txt.

In addition to text which needs to be translated, input documents will likely contain additional data such as formatting information, formatting tags, images, etc. The File Filtering process involves identifying the (textual) localizable content. This content is extracted and decoupled from any non-translatable content (the exception are in-line formatting tags, such as the ones used to indicate italics or boldface, which are preserved and encapsulated), resulting in a clean text version of the content which is ready to be translated, and which a linguist can edit without needing to purchase a license for the software the original document was saved in, e.g. Microsoft Office.

In addition to producing an XLIFF file, the File Filtering module also produces a skeleton of the document which contains information complementary to that in the XLIFF and which is needed to rebuild the original file format. This is used in the last stage of the workflow to produce a final document which has the same formatting as the original, but where the content has been translated.

3.2 Translation Memory

Users can upload TM files containing their previously translated data. SmartMATE is able to import TMs stored in the standard TMX[4] (Translation Memory eXchange) format, which can be exported from any Translation Management System software.

TMs inside SmartMATE can be exploited in two different ways. Firstly, they can be used as traditional Translation Memories. When a new document is ready for translation, any segment in the document which exactly matches a TM entry will appear in the editor suite as pre-translated using the target side of this entry. In addition to exact matches, SmartMATE also leverages entries which only match above a predefined match threshold (Fuzzy Matches) (Sikes, 2007), and is able to identify In-Context Exact (ICE) matches, i.e. segments which are an exact match and which are preceded and followed by an exact match segment. After a document has been translated and signed-off by the proofreader, TMs can be automatically updated to include the newly translated content.

In addition to being used as traditional TMs, any TMX uploaded by the user can be used to train an MT engine, as explained in the following section.

3.3 Machine Translation

After TM files have been uploaded, these can be used to train MT engines. After the user has completed a simple form with the details of their requested engine, a process starts which requires no human intervention and which produces a state-of-the-art SMT engine. The process begins by extracting plain bilingual text from the TMX files, thus creating a parallel corpus. This is then subject to multiple stages of corpus cleaning which include:

- ensuring the correct character encodings are being used,
- removing any formatting tags so that they do not interfere with the training process,
- removing duplicate sentence pairs,
- removing sentence pairs which exceed certain source:target length ratio,
- replacing entities such as URLs and e-mails with placeholders to improve the generalization of the statistical models.

After the corpus has been cleaned, 1,000 randomly selected sentence pairs are kept apart for

[4] http://www.gala-global.org/oscarStandards/tmx/tmx14b.html

evaluation purposes, and an additional 500 sentence pairs for tuning. The remaining data is used to train SMT models using the Moses (Koehn et al., 2007) toolkit. The user is then presented with the built engine along with automatically obtained BLEU (Papineni et al., 2002) scores, which are calculated over the 1,000 randomly held-out sentence pairs and which give an indication of the level of translation quality that could be expected from this engine when used to translate documents of a nature similar to those used when training the engine.

The process of building an engine involves creating phrase-based translation models (Koehn et al., 2003) and lexicalized reordering models (Koehn et al., 2005) as well as a Language Model (LM), for which the IRSTLM toolkit (Federico and Cettolo, 2007) is used. In addition, the model weights are optimized using Minimum Error Rate Training (Och, 2003) so as to maximize the BLEU score over the 500 sentence pairs randomly held out from the original TMs for tuning. All of this complexity, as well as the significant hardware requirements needed to host the engine training, are hidden from the user.

It is worth noting that since these engines have been built using the user's own data, they are specialized engines from which a better translation quality can be expected[5] when compared to general-purpose engines such as those provided by services such as Google Translate[6] or Microsoft Bing Translator,[7] which in addition might not offer the same data privacy guarantees as SmartMATE.

3.4 Terminology

SmartMATE is able to import multilingual glossaries containing user-specific terminology. The accepted formats are CSV (Comma-Separated Values) files, which are obtainable from any spreadsheet software, or the standard TBX (TermBase eXchange) (ISO 30042, 2008).

These glossaries can be exploited in several ways. Firstly they can be used as a complement of TMX files during MT engine building. This has the effect of improving word alignment (and subsequently phrase-alignment), as it provides reference points for the SMT alignment algorithms (Och and Ney, 2000). Secondly, they can be used for glossary-injection during MT. Once an engine has been trained, glossaries can be used while the engine is processing an input document to ensure that the MT output adheres to the terminology specified by the glossary. When using multiple glossaries which provide conflicting entries for the same source term, all of the possible target translations are provided to the engine, which uses its LM to determine which translation option provides the most fluent target sentence.

Finally, the editor suite supports the use of glossaries as well, by highlighting any source term which matches a source segment, and providing to the linguist the available target terms. The editor is also able to detect whether the target term specified in the glossary has been used in translating the segment, and to flag with a warning segments which do not conform to entries in the glossary.

3.5 Editor Suite

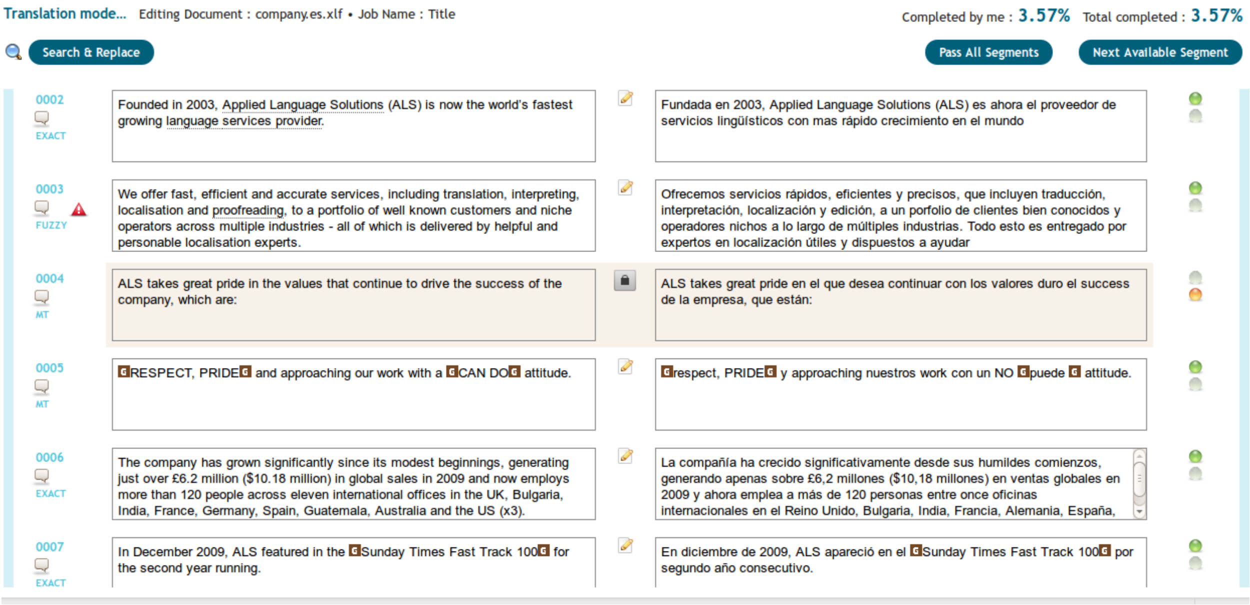
The editor suite integrates all of SmartMATE's capabilities, effectively providing the user with a single tool that can be used for the complete translation workflow. SmartMATE is cloud-based, as it is hosted on Amazon's cloud. This has several beneficial implications. Firstly, data is automatically saved at segment level, which means that any technical problem on the user's computer will not affect the integrity of the translated data. Secondly, the user is able to access their data from any computer which is equipped with an internet connection. Even though a collection of TMs and MT engines can easily require several Giga Bytes of disk space to be stored, the user can quickly access this data from any computer with an internet browser. Finally, its cloud-based nature means that SmartMATE is able to scale virtually arbitrarily. Regardless of the amount of users currently accessing the system or running MT engines, each user is assigned a dedicated virtual PC in the cloud so that system performance is unaffected.

The editor provides two operation modes: translation and proofreading, which we discuss in the following sections.

[5] This is mainly due to the ambiguity introduced by out-of-domain data (Sennrich, 2012), and is a known effect in the domain adaptation literature, e.g. (Foster et al., 2010)

[6] http://translate.google.com

[7] http://www.microsofttranslator.com

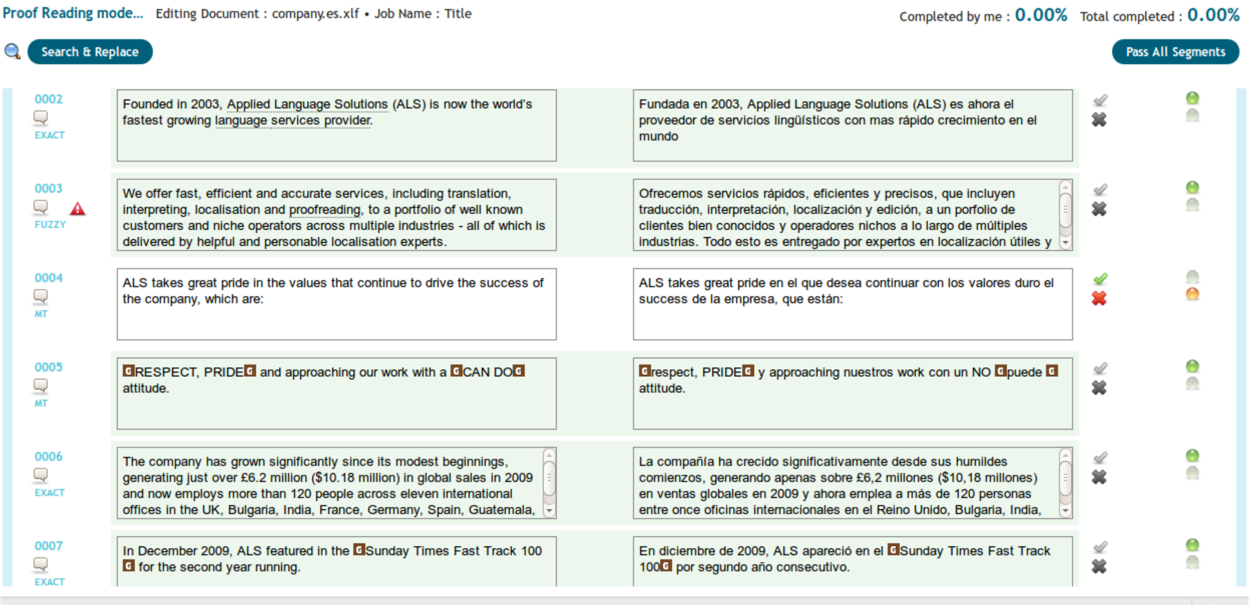


Translation mode... Editing Document : company.es.xlf • Job Name : Title

Completed by me : 3.57% Total completed : 3.57%

0002 EXACT	Founded in 2003, Applied Language Solutions (ALS) is now the world's fastest growing language services provider.	Fundada en 2003, Applied Language Solutions (ALS) es ahora el proveedor de servicios lingüísticos con mas rápido crecimiento en el mundo
0003 FUZZY	We offer fast, efficient and accurate services, including translation, interpreting, localisation and proofreading, to a portfolio of well known customers and niche operators across multiple industries - all of which is delivered by helpful and personable localisation experts.	Ofrecemos servicios rápidos, eficientes y precisos, que incluyen traducción, interpretación, localización y edición, a un porfolio de clientes bien conocidos y operadores nichos a lo largo de múltiples industrias. Todo esto es entregado por expertos en localización útiles y dispuestos a ayudar
0004 MT	ALS takes great pride in the values that continue to drive the success of the company, which are:	ALS takes great pride en el que desea continuar con los valores duro el success de la empresa, que están:
0005 MT	GRESPECT, PRIDEG and approaching our work with a GCAN DOG attitude.	Grespect, PRIDEG y approaching nuestros work con un NO Gpuede G attitude.
0006 EXACT	The company has grown significantly since its modest beginnings, generating just over £6.2 million ($10.18 million) in global sales in 2009 and now employs more than 120 people across eleven international offices in the UK, Bulgaria, India, France, Germany, Spain, Guatemala, Australia and the US (x3).	La compañía ha crecido significativamente desde sus humildes comienzos, generando apenas sobre £6,2 millones ($10,18 millones) en ventas globales en 2009 y ahora emplea a más de 120 personas entre once oficinas internacionales en el Reino Unido, Bulgaria, India, Francia, Alemania, España,
0007 EXACT	In December 2009, ALS featured in the GSunday Times Fast Track 100G for the second year running.	En diciembre de 2009, ALS apareció en el GSunday Times Fast Track 100G por segundo año consecutivo.

Figure 2: Translation mode in the editing environment



Proof Reading mode... Editing Document : company.es.xlf • Job Name : Title

Completed by me : 0.00% Total completed : 0.00%

0002 EXACT	Founded in 2003, Applied Language Solutions (ALS) is now the world's fastest growing language services provider.	Fundada en 2003, Applied Language Solutions (ALS) es ahora el proveedor de servicios lingüísticos con mas rápido crecimiento en el mundo
0003 FUZZY	We offer fast, efficient and accurate services, including translation, interpreting, localisation and proofreading, to a portfolio of well known customers and niche operators across multiple industries - all of which is delivered by helpful and personable localisation experts.	Ofrecemos servicios rápidos, eficientes y precisos, que incluyen traducción, interpretación, localización y edición, a un porfolio de clientes bien conocidos y operadores nichos a lo largo de múltiples industrias. Todo esto es entregado por expertos en localización útiles y
0004 MT	ALS takes great pride in the values that continue to drive the success of the company, which are:	ALS takes great pride en el que desea continuar con los valores duro el success de la empresa, que están:
0005 MT	GRESPECT, PRIDEG and approaching our work with a GCAN DOG attitude.	Grespect, PRIDEG y approaching nuestros work con un NO Gpuede G attitude.
0006 EXACT	The company has grown significantly since its modest beginnings, generating just over £6.2 million ($10.18 million) in global sales in 2009 and now employs more than 120 people across eleven international offices in the UK, Bulgaria, India, France, Germany, Spain, Guatemala,	La compañía ha crecido significativamente desde sus humildes comienzos, generando apenas sobre £6,2 millones ($10,18 millones) en ventas globales en 2009 y ahora emplea a más de 120 personas entre once oficinas internacionales en el Reino Unido, Bulgaria, India,
0007 EXACT	In December 2009, ALS featured in the GSunday Times Fast Track 100 G for the second year running.	En diciembre de 2009, ALS apareció en el GSunday Times Fast Track 100G por segundo año consecutivo.

Figure 3: Proofreading mode in the editing environment

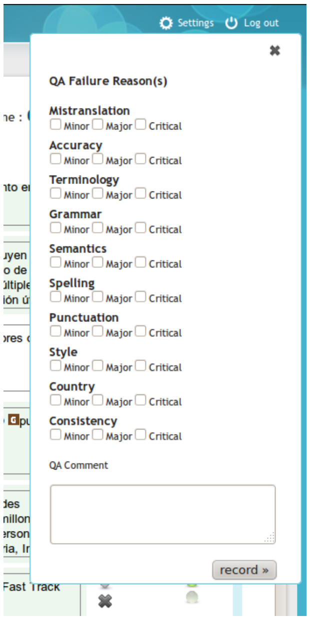
3.5.1 Translation

Figure 2 shows SmartMATE's editor suite in translation mode. There are two main columns, with the left one showing the translatable source content which was extracted from the original file, and the right one the corresponding target segments. Depending on which modules were activated by the user, the initial content in the target segments will change. In this particular example, both TM and MT were activated, as can be observed from the information displayed to the left of each segment. Segments are labelled according to whether they resulted in a TM match (either exact, fuzzy or in-context exact), or whether they were sent to MT.

This figure also illustrates the use of glossaries within the editor. Segments 2 and 3 contain source terms which have been highlighted, meaning that these terms matched a glossary entry. Hovering the mouse over these terms will show the translations suggested by the glossary. In addition, when editing the target side of a segment, linguists have access to a Glossary tab from which they can easily incorporate glossary terms into the translation. The red warning sign in segment 3 illustrates how SmartMATE indicates that a segment contains glossary matches but the target terms specified in the glossary have not been used in the translation.

Once a translator has finished editing a segment, the segment can be locked. This is automatically done by the Editor when switching to a different segment, or can be explicitly triggered by clicking on the dedicated button which separates source from target segments. In Figure 2, only segment 4 has been locked, which is indicated by a different background colour and a lock symbol. When a segment is locked, it instantly becomes available for the next stage of the workflow, e.g. proofreading. See Section 3.5.2 for the concurrency implications of being able to lock an individual segment, rather than the complete document.

Finally, segment 5 shows how in-line formatting can be protected. In the original file, the words "RESPECT, PRIDE" were typed in boldface. SmartMATE's editor hides this formatting to the user, but explicitly shows that there is formatting information which should be preserved. Linguists can drag and drop these protected tags from source to target so as to keep the formatting. The same principle can be applied to preserve tags when translating structured documents such as HTML or XML files.



QA Failure Reason(s)

Mistranslation
☐ Minor ☐ Major ☐ Critical

Accuracy
☐ Minor ☐ Major ☐ Critical

Terminology
☐ Minor ☐ Major ☐ Critical

Grammar
☐ Minor ☐ Major ☐ Critical

Semantics
☐ Minor ☐ Major ☐ Critical

Spelling
☐ Minor ☐ Major ☐ Critical

Punctuation
☐ Minor ☐ Major ☐ Critical

Style
☐ Minor ☐ Major ☐ Critical

Country
☐ Minor ☐ Major ☐ Critical

Consistency
☐ Minor ☐ Major ☐ Critical

QA Comment

Figure 4: LISA QA-compliant feedback form

3.5.2 Proofreading

In addition to allowing the post-editing of MT output (and/or fuzzy TM matches, depending on which modules were activated for a particular job), SmartMATE also supports a proofreading stage were a different linguist can asses the work done by the translators, ensuring the coherence of the complete document, the adherence to client-specific policies and terminology, etc.

Figure 3 shows the proofreader's perspective of the document which is being translated in Figure 2. As can be seen, only segment 4 has become available for proofreading, as this is the only segment which has so far been locked by the translator.

Proofreaders are able to edit the target segments, and mark each segment as finished. If a translated segment contains severe errors, the proofreader can send the segment back to the translation phase, by clicking on the red cross next to it. When doing so, they can record detailed information about the linguist's reasons why the segment has been rejected, by using the form shown in Figure 4. This form conforms to the Localization Industry Standards Association (LISA) QA Model.

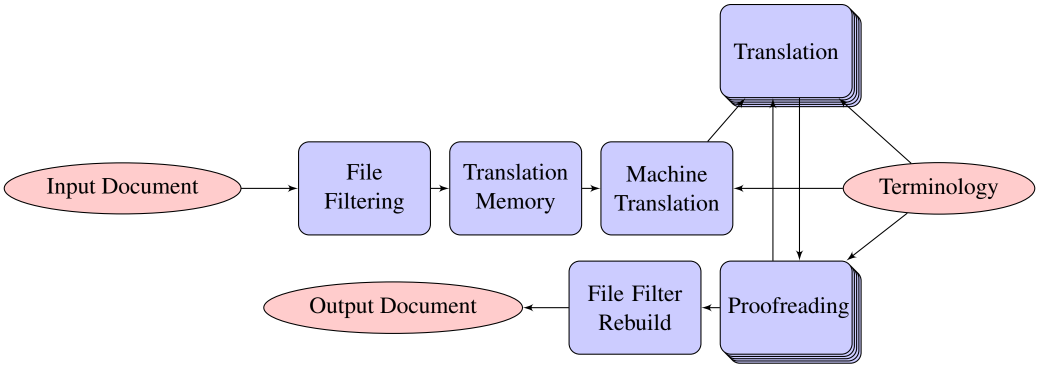


Figure 5: Possible translation workflow in SmartMATE

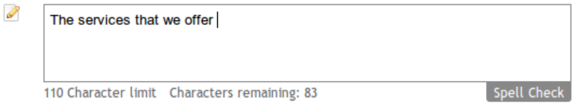

The services that we offer

Figure 6: Character length limit being enforced to a segment by the Editor

Note that content becomes available for proofreading at *segment* level. That is, as soon as a translator has locked a segment, the proofreader is able to post-edit it and send it to the next stage, or send it back to translation. This means that, in addition to supporting the traditional (linear) workflow of Figure 1, the editor in SmartMATE enables proofreading to be done simultaneously to translation, effectively reducing proofreading time to zero. While some projects are best suited by the traditional linear workflow, there certainly are situations in which this concurrency model is desirable. In effect, SmartMATE allows a workflow such as the one in Figure 5, where translation and proofreading run concurrently. Additionally, SmartMATE allows multiple users to collaborate on the same document at the same time, enabling further reductions in translation time.

4 Case Study

In order to demonstrate the robustness and usefulness of our tool, we discuss in this section a translation project which is being carried out for Spil Games,[8] a large online games developer and publisher of the type seen on social networking sites. Games are originally written in English, and are subsequently localized into over 15 languages for a global audience of more than 180 million monthly active users.

Spil Games provides the localizable content to the author's institution (ALS), which is in charge of File Engineering, Project Management, TM/MT application and translation. Reviewing, however, is outsourced to a third party (VistaTEC).[9] The whole process is supported by and hosted in SmartMATE. ALS creates a new translation job in SmartMATE, and assigns the reviewing task to VistaTEC. Once the translation stage is complete, VistaTEC can itself delegate the reviewing to an arbitrary number of SmartMATE users from within the tool. The identity of the linguists who review the content is not revealed to ALS, thus ensuring VistaTEC's commercial confidentiality.

During the first stages of the project, only TM and Glossaries are used. However, after each new document has been translated, SmartMATE automatically updates the Translation Memories so that this newly created content can be matched against future documents. During the course of the project, as more content is translated the TM files will eventually reach a size substantial enough to allow customer-specific engines to be trained from them. We expect significant improvements in translation speed to be achieved once this happens.

The content translated for company A must satisfactorily be displayed inside the User Interface of a game, which means that some segments must conform to length restrictions. This requirement is ac-

[8] http://www.spilgames.com/

[9] http://www.vistatec.com/

Target Language	Segments	Source Words	Target Words	Exact	Fuzzy
Portuguese (Brazilian)	262	3,997	4,110	24%	6%
Russian	257	3,810	3,294	25%	6%
Turkish	250	3,608	3,183	24%	7%
Indonesian	256	3,787	3,327	24%	6%
Dutch	295	4,286	3,728	24%	5%
Portuguese (Portugal)	211	2,663	2,866	28%	8%
German	264	3,951	3,869	23%	6%
French	242	3,538	3,845	22%	5%
Swedish	289	4,089	3,923	21%	6%
Spanish	258	3,914	4,344	24%	6%
Italian	208	2,796	3,083	30%	6%
Polish	238	3,059	2,944	26%	7%
Arabic (Modern Standard)	111	2,353	1,851	0%	0%

Table 1: Statistics for each language pair in the project

commodated in SmartMATE by allowing a character limit to be specified in an XLIFF element at segment level, using the `maxwidth` property. Spil Games can then specify the desired limit, and this is enforced by the editor, as illustrated in Figure 6.

We give in Table 1 statistics gathered during one of the first weeks in the project. During this week, an average of 241 segments were translated from English into 13 language pairs, which amount to 45,851 source words among all language pairs. Although the average sentence length among all of the English segments is 14.6 words, there is a large variance. Most of the content to be translated consists of titles and descriptions. Titles tend to be quite short, while descriptions are longer. We see that for most language pairs, an exact match rate of between 20% and 30% is achieved. Although this means that a significant amount of translation work is reduced due to SmartMATE exploiting our customer's TMs, we noticed that most of the matching segments were titles rather than descriptions. We expect, however, that as TMs grow in size, a larger number of long segments will be able to be matched, and that the incorporation of post-edited MT into the project will significantly reduce turn-around times.

5 Conclusions and Future Work

In this paper we have presented SmartMATE, an online self-serve MT translation platform, which integrates TM, MT and Terminology into a powerful editing environment. We have shown not only how the complete localisation workflow can be accommodated using this single tool, but also how the concurrency capabilities of the editor enable additional workflows to be considered. In addition we have studied the first stages of a particular project from a large client which is currently being run using SmartMATE, showing that our product is robust enough to be used in large-scale production environments. We believe that SmartMATE has the capability of empowering non-technical users with MT technology, and of advancing the standards in the localisation industry.

There are many areas in which we can continue to improve SmartMATE. In the short term, we will focus on extending the number of file formats supported by our file filtering module (e.g. pdf), and on enabling advanced modules when training MT engines, such as named entity recognizers, segmenters, tokenizers and compound splitters.

Acknowledgments

Thanks to Laura van Nigtevegt from Spil Games for reviewing the paper and for her helpful comments. We also thank the anonymous reviewers for their comments and suggestions.

References

Wilker Aziz, Sheila Castilho Monteiro de Sousa, and Lucia Specia. 2012. PET: a tool for post-editing and assessing machine translation. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 3982–3987, Istanbul, Turkey.

Donald A. DePalma and Nataly Kelly. 2009. The business case for machine translation. Common Sense Advisory. `http://www.commonsenseadvisory.com/AbstractView.aspx?ArticleID=859`.

Marcello Federico and Mauro Cettolo. 2007. Efficient Handling of N-gram Language Models for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 88–95, Prague, Czech Republic.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459, Cambridge, MA.

ISO 30042. 2008. *Systems to manage terminology, knowledge and content – TermBase eXchange (TBX)*. ISO, Geneva, Switzerland.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–52, Edmonton, Canada.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation*, Pittsburgh, PA.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Annual Meeting of the ACL, demonstation session*, pages 177–180, Prague, Czech Republic.

Franz Josef Och and Hermann Ney. 2000. A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th conference on Computational linguistics - Volume 2*, pages 1086–1090, Saarbrücken, Germany.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.

Rico Sennrich. 2012. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 539–549, Avignon, France.

Richard Sikes. 2007. Fuzzy matching in theory and practice. *Multilingual*, 18(6):39–43.

Andy Way, Kenny Holden, Lee Ball, and Gavin Wheeldon. 2011. SmartMATE: Online self-serve access to state-of-the-art SMT. In *Proceedings of the Third Joint EM+/CNGL Workshop "Bringing MT to the User: Research Meets Translators", JEC 2011*, pages 43–52, Luxembourg.

To post-edit or not to post-edit?

Estimating the benefits of MT post-editing for a European organization

Alexandros Poulis
Intrasoft Intl
European Parliament
alexandros.poulis@ext.europarl.europa.eu

David Kolovratnik
Charles Oakes & Co
European Parliament
david.kolovratnik@ext.europarl.europa.eu

Abstract

In the last few years the European Parliament has witnessed a significant increase in translation demand. Although Translation Memory (TM) tools, terminology databases and bilingual concordancers have provided significant leverage in terms of quality and productivity the European Parliament is in need for advanced language technology to keep facing successfully the challenge of multilingualism. This paper describes an ongoing large-scale machine translation post-editing evaluation campaign the purpose of which is to estimate the business benefits from the use of machine translation for the European Parliament. This paper focuses mainly on the design, the methodology and the tools used by the evaluators but it also presents some preliminary results for the following language pairs: Polish-English, Danish-English, Lithuanian-English, English-German and English-French.

1 Introduction

The European Parliament (EP) has witnessed a significant increase in translation requests in the last few years. For instance the total amount of source pages translated by the Directorate General for Translation (DGTRAD) in the first quarter of 2010 was 43,963. In the first quarter of 2012 this number increased to 60,275 while the number of translators has remained rather stable. This situation requires a significant productivity increase in the most cost-efficient way so that DGTRAD can keep accomplishing its mission: making available in all official languages of the European Union (EU) all documents relating to EP's role as co-legislator and enabling the EP to permit all EU citizens to communicate with the EU institutions in their own language as efficiently and effectively as possible.

So far all this has been possible thanks to the extensive use of various translation technologies such as Translation Memory systems, terminology databases, bilingual concordancers and other reference tools which have been seamlessly integrated in the translation workflow in the last 6 years. Nevertheless, current demand requires new technologies to be tested and Machine Translation is probably the most important one.

To examine what can be expected and evaluate the most obvious deficiencies we organized a large-scale evaluation of a general-purpose MT system developed by the European Commission (Eisele et al. 2011). The tests will be conducted by 62 translators in 24 language pairs.

1.1 Use-case

The vast majority of EP documents are written in English, with French and German following in the second and third place. On that basis we decided to start testing the following language pairs: English to all official EU languages (Table 1), German to English and French to English. Each evaluator works always from one source language into her mother tongue.

For the current round of tests we have selected documents which do not contain highly repetitive text and therefore their segments are rarely found in our translation memories. Some of these document types are parliamentary questions, petitions, notes from various bodies of the EP and draft resolutions[1]. With translation memories not providing much input for those document types we see a strong case where MT could be of some help to translators.

MT can and most probably will be used for other purposes such as communication and gisting but this study focuses only on its use as a translation aid.

EU Languages	
Bulgarian	Italian
Czech	Latvian
Danish	Lithuanian
Dutch	Maltese
English	Polish
Estonian	Portuguese
Finnish	Romanian
French	Slovak
German	Slovene
Greek	Spanish
Hungarian	Swedish
Irish	

Table 1: EU official languages.

2 Translation technologies in the current workflow

The current translation workflow relies largely on Translation Memory (TM) technology which is the main component of the so called Translation Environment Tools (TEnT). TMs are large databases that contain pairs of segments (usually sentences) in the source and target language. Each such pair of segments is called a translation unit. Translation memories can be bilingual (one source and one target language) or multilingual (one source and multiple target languages). In the EP the available TEnTs support only bilingual memories although this will change in the near future. As one source segment may have more than one translation equivalents within the same TM, each translation unit contains also some meta-data that provide information about its origin, creator, requestor and its creation date. These meta-data can help the translator assess the reliability of each available translation option for a given segment and select the most appropriate translation in a given context.

While translating a document each source segment is compared to the TM content and translations of matching segments are proposed by the system. Matching segments can be either identical to the source segment (100% or full match) or similar to it (fuzzy match). Fuzzy matches are usually between 65% and 99%. Full matches are usually accepted without changes but fuzzy matches need to be post-edited.

Besides TMs our translators have access to large terminology databases which are constantly enriched with the support of a dedicated terminology service which makes sure that the terminological entries are inserted in time for new translation projects and that they are complete including all of our working languages and references following expert translators' or terminologists' quality approval.

Bilingual concordancers enable searches of terms, phrases or any strings within their context. Depending on the input format of the tool that context can be a whole document or just a translation memory segment.

An interinstitutional search engine called Quest2 brings many databases under a common user interface and offers almost 4,500 translators access to various reliable terminology, document and TM resources.

All these tools have helped the EP cope with the increasing workload so far. Nevertheless, it is clear that some additional leverage is needed and MT seems to be the way to go. In section 2 we mentioned that TMs can provide 100% matches and fuzzy matches. If no matching segment is found in the database or if the match value is lower than 65% the source segment needs to be translated from scratch. It is primarily –but not exclusively- in those cases that MT can be of use if it is of sufficient quality to allow for faster post-editing than translating the whole segment. When fuzzy matches are available the MT output will be offered to the user too. Previous research such as (Simard et al. 2009) has shown that MT performs better when there is also a good fuzzy match and its usability may even outperform that of the corresponding fuzzy match. In-house experience has shown that MT output can help translators edit the fuzzy matches faster. Taking this into consideration we are currently investigating the possibility of automatically enhancing the fuzzy

[1] For more information about and access to European Parliament's documents please visit http://www.europarl.europa.eu/RegistreWeb/search/typedoc.htm?language=EN

matches with MT implementing the algorithm proposed by (Koehn et al. 2010). The introduction of MT to the workflow will have a great impact on the role of translators. They will now mainly be asked to post-edit TM and MT output rather than translate free text at least as far as certain document types and language pairs are concerned.

3 The project MT@EP

Following the promising results of the Exodus experiment which was presented in (Jellinghaus et al. 2010) the EP DGTRAD decided to launch an IT project with the objective of estimating the benefits of MT and ensuring its efficient implementation in the translation workflow but also potentially as a communication tool between staff members or between the citizens and the EP. This paper focuses only on the first use case - MT as a translation aid. MT is expected to bring certain benefits to the EP; therefore MT post-editing is being carefully evaluated taking into account various parameters which are presented in section 4.

3.1 Expected business benefits

DGTRAD expects that MT will help increase its translation productivity - measured in number of standard pages[2] per period of time - at least for certain document categories/domains and language pairs. MT is expected to offer more added value to domains with higher availability of internal documents that can be used in the training corpora as well as to language pairs with higher data availability and similarity between source and target. First experiments confirm this view showing that reaching usable MT quality levels when translating into Finish, Hungarian or other morphologically rich languages is much more challenging than most other language pairs. This does not come as a surprise as it has been repeatedly observed in the MT literature as for example in (Koehn 2005). To what extent can DGTRAD expect MT to increase its translation productivity and how can we estimate that? This is the main question that we will try to answer in the next sections of this paper using a MT post-editing and some other MT evaluation tasks.

At the same time it is expected that MT will help maintain a high level of translation quality by helping translators cope with their workload in the available amounts of time. The continuous increase of translation requests could, in theory, have an impact on the quality of translated documents. Nevertheless, this cannot be allowed for legislative documents as it will most certainly affect the whole legislative procedure.

Furthermore, MT may contribute to a better value for money of translations particularly by reducing the cost of translation outsourcing per outsourced page. The overall expenses for external translations may not decrease but possibly lower charges for machine translated segments may provide an opportunity for more documents to be outsourced.

Unlike TM, Machine Translation does not include references to the source of translations. TM meta-data indicate which document a proposed translation comes from, which legislative procedure it is linked to, when it was produced etc. The lack of this information in our current MT implementation will have an impact on post-editing time even if the MT output is linguistically perfect in particular in the case of legislative documents. This is mainly due to the fact that our translators are obliged to re-use the exact same translations that have been produced in other documents which are being referred to in the current source document. If the source of a machine translated string is unknown then the translators will have to spend some time controlling the origin of certain translation suggestions and this is a risk with a direct impact on the above mentioned expected benefits.

3.2 Project deliverables

The main deliverable of this project is an MT solution for more than 700 in-house translators and 506 language pairs - from and into all EU official languages. The quality of the MT is expected to be good enough to reduce translation time in all language combinations while there is also a use case for raw MT for gisting purposes (without or with minimal revision). In this case the MT output is expected to be of understandable but not necessarily of human quality.

Synchronous (real-time) MT services are currently out of this project's scope. Machine translated segments will be incorporated in the translation memories and offered as part of a pre-

[2] One "standard page" consists of 1500 characters

translation package. Pre-translation packages are prepared and provided to translators before the beginning of a translation task and nowadays they usually include translation memory segments relevant to their working document. Real-time MT would require a significantly higher investment on hardware resources to achieve much faster decoding times.

3.3 Data

European Institutions have established a close collaboration framework in the area of translation technologies. The first step was taken with Euramis (Blatt 1998), a huge translation memory with almost 300 million segments available to different EU institutions. Thanks to Euramis the Council, the Court of Auditors, the Court of Justice, the Committee of the Regions, the European Economic and Social Committee, the Parliament and the Translation Centre for the Bodies of the European Union can contribute to each others work by adding their own translation segments.

Along with translation memories the EP has also important amounts of documents in its archives as well as on its web-site many of which are not included in the translation memories. These resources are being collected and parallelised to be used for MT purposes. In the future external corpora that have not been produced in-house should also be incorporated.

4 MT evaluation

To estimate the expected benefits described in section 3.1 the EP is conducting a large scale MT evaluation for the first time in its history relying on the contribution of 62 in-house translators. The main conclusions we expect to draw concern MT quality, MT comprehensibility and MT editing time compared to translation time. The test users work with a web-based evaluation tool which was initially used for ACL's WMT workshop and described in (Callison-Burch et al. 2009) and configured in-house to meet our own specifications. The MT solution that is being tested at this stage is the one developed by the European Commission which is described in (Eisele et al. 2011). This solution has been chosen in the context of interinstitutional collaboration which started in 2009 in the MT field and it is a statistical MT system based on Moses (Koehn et al. 2007).

4.1 Methodology

For the selection of the evaluation methodology the MT@EP project team has collaborated with a user group that has been created for this purpose. The participants of the user group are mainly representatives of the business (translators), one business analyst and one computational linguist with many years of experience in MT.

First of all the document types were carefully selected as MT seems to be more appropriate for some than for the others. Legislative documents were left out of this process because lacking the source documents of MT-translated strings translators would not be able to evaluate or post-edit the MT output as required by the testing specifications. Therefore documents with more free text, less quotes and of diverse domains and language registers were chosen.

Translation demand was another parameter that was taken into consideration when the test corpus was selected. Therefore, document types more frequently translated than others have been selected.

4.1.1 Categorization and Error Detection

To evaluate MT quality the test users are provided with segments in the source language and their machine translated equivalents and they are asked to mark them as Excellent, Good, Medium or Poor. Test instructions provide precise definitions of those marks to make sure that the test users take common criteria into consideration to the extent that this is possible. Here we used the categories used by (Roturier 2009). More precisely the test users were provided with the following definitions:

Excellent MT Output: Your understanding is not improved by the reading of the source because it is syntactically correct; it uses proper terminology; the translation conveys information accurately.

Effect: No post-editing required.

Good MT Output: Your understanding is not improved by the reading of the source even though the MT segment contains minor errors affecting any of these: grammatical (article, preposition), syntax (word order), punctuation, word formation (verb endings, number agreement), unacceptable style. An end-user who does not have access to the source text could possibly understand the MT segment.

Effect: Only minor post-editing required in terms of actual changes or time spent post-editing.

Medium MT Output: Your understanding is improved by the reading of the source, due to significant errors in the MT segment (textual coherence / textual pragmatics / word formation / morphology). You would have to re-read the source text a few times to correct these errors in the MT segment. An end-user who does not have access to the source text could only get the gist of the MT segment.

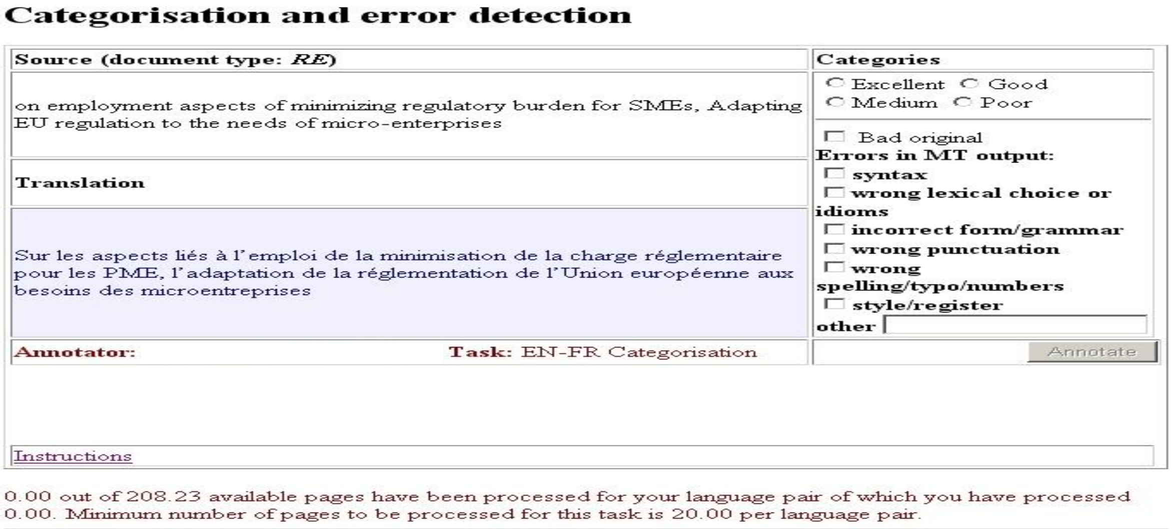
Effect: Severe post-editing is required or maybe just minor post-editing after spending too much time trying to understand the intended meaning and where the errors are.

Poor MT Output: Your understanding only derives from the reading of the source text, as you could not understand the MT segment. It contained serious errors in any of the categories listed above, including wrong Parts Of Speech. You could only produce a translation by dismissing most of the MT segment and/or re-translating from scratch. An end-user who does not have access to the source text would not be able to understand the MT segment at all.

Effect: It would be better to manually retranslate from scratch (post-editing is not worthwhile). Moreover the participants have the option of selecting among some basic types of errors in the MT: syntax, wrong lexical choice or idioms, incorrect form/grammar, wrong punctuation, wrong spelling/typo/numbers, style/register. We didn't provide a more detailed error classification because at this stage we prefer receiving more evaluation data than feedback on specific error types.

There can be cases where the MT output is fluent but it is not clear to the translator if it conveys the message of the original text for the simple reason that often the original text may be incomprehensible (badly formulated or out of context). Therefore, the test-users are able to mark a bad original as such.

To evaluate the comprehensibility of MT and its appropriateness for gisting purposes a next task offers the test users a paragraph in the source language with its MT target. In this task test users only need to state if the translation conveys the meaning of the original text or not. They are also given the option to select "bad original". To make sure that users would not abuse the latter in order to proceed to the next segment they still have to state if the MT output conveys the meaning of the original text instead of proceeding directly to the next one. In the opposite case test-users may feel tempted to skip the most complicated cases or paragraphs containing long sentences. This is the only task where paragraphs are provided instead of segments because context is often necessary to understand the information contained in a single sentence.



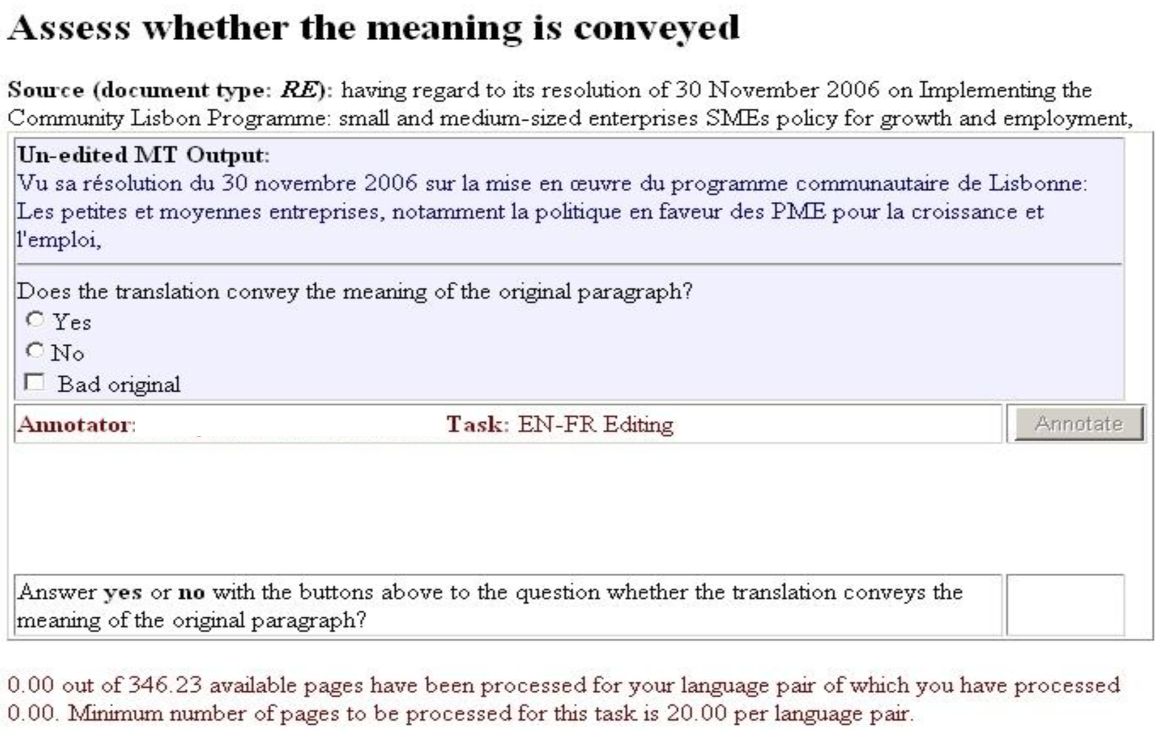
Categorisation and error detection

Source (document type: *RE*)	**Categories**
on employment aspects of minimizing regulatory burden for SMEs, Adapting EU regulation to the needs of micro-enterprises	○ Excellent ○ Good ○ Medium ○ Poor
Translation	□ Bad original **Errors in MT output:** □ **syntax** □ **wrong lexical choice or idioms** □ **incorrect form/grammar** □ **wrong punctuation** □ **wrong spelling/typo/numbers** □ **style/register** **other**
Sur les aspects liés à l'emploi de la minimisation de la charge réglementaire pour les PME, l'adaptation de la réglementation de l'Union européenne aux besoins des microentreprises	
Annotator: **Task:** EN-FR Categorisation	Annotate

Instructions

0.00 out of 208.23 available pages have been processed for your language pair of which you have processed 0.00. Minimum number of pages to be processed for this task is 20.00 per language pair.

Figure 1: Categorization and error detection task



Assess whether the meaning is conveyed

Source (document type: *RE*): having regard to its resolution of 30 November 2006 on Implementing the Community Lisbon Programme: small and medium-sized enterprises SMEs policy for growth and employment,

Un-edited MT Output:
Vu sa résolution du 30 novembre 2006 sur la mise en œuvre du programme communautaire de Lisbonne: Les petites et moyennes entreprises, notamment la politique en faveur des PME pour la croissance et l'emploi,

Does the translation convey the meaning of the original paragraph?
○ Yes
○ No
□ Bad original

Annotator: **Task:** EN-FR Editing

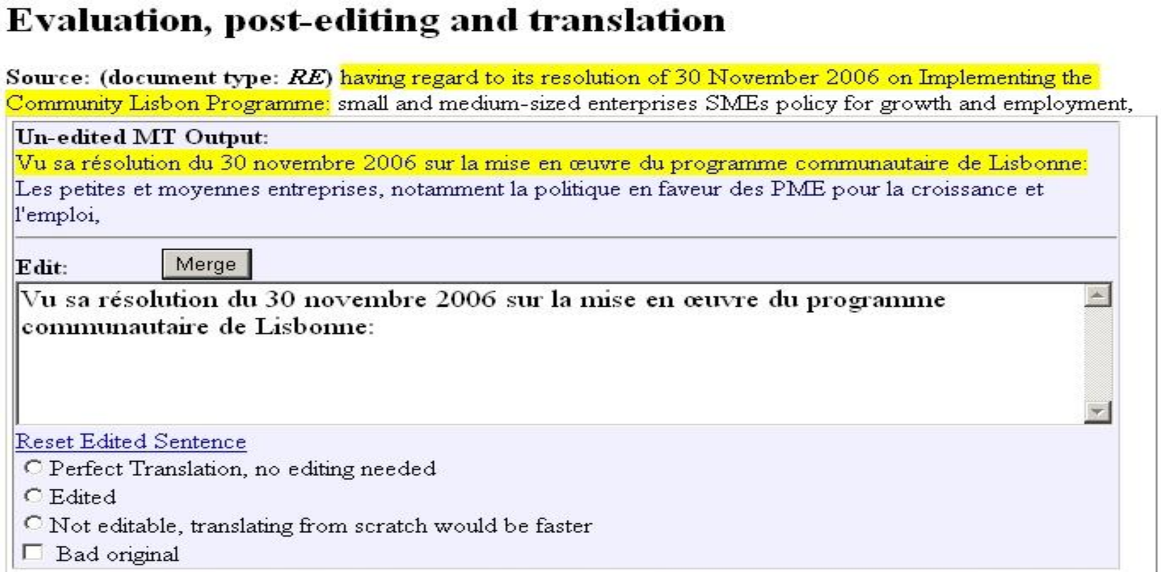
Answer **yes** or **no** with the buttons above to the question whether the translation conveys the meaning of the original paragraph?

0.00 out of 346.23 available pages have been processed for your language pair of which you have processed 0.00. Minimum number of pages to be processed for this task is 20.00 per language pair.

Figure 2: Paragraph assessment for gisting purposes

4.1.2 Post-Editing and Translation

Approximately 80% of the paragraphs displayed in the previous task are machine-translated. The purpose of the post-editing task is to edit the MT output until it's considered to be of publishable quality. If the MT output is already of publishable quality users select "Perfect Translation, no editing needed". The post-editing time is measured from the moment that the page is loaded until the end of the last action taken (editing, selection of a radio-button etc.).



Evaluation, post-editing and translation

Source: (document type: *RE*) having regard to its resolution of 30 November 2006 on Implementing the Community Lisbon Programme: small and medium-sized enterprises SMEs policy for growth and employment,

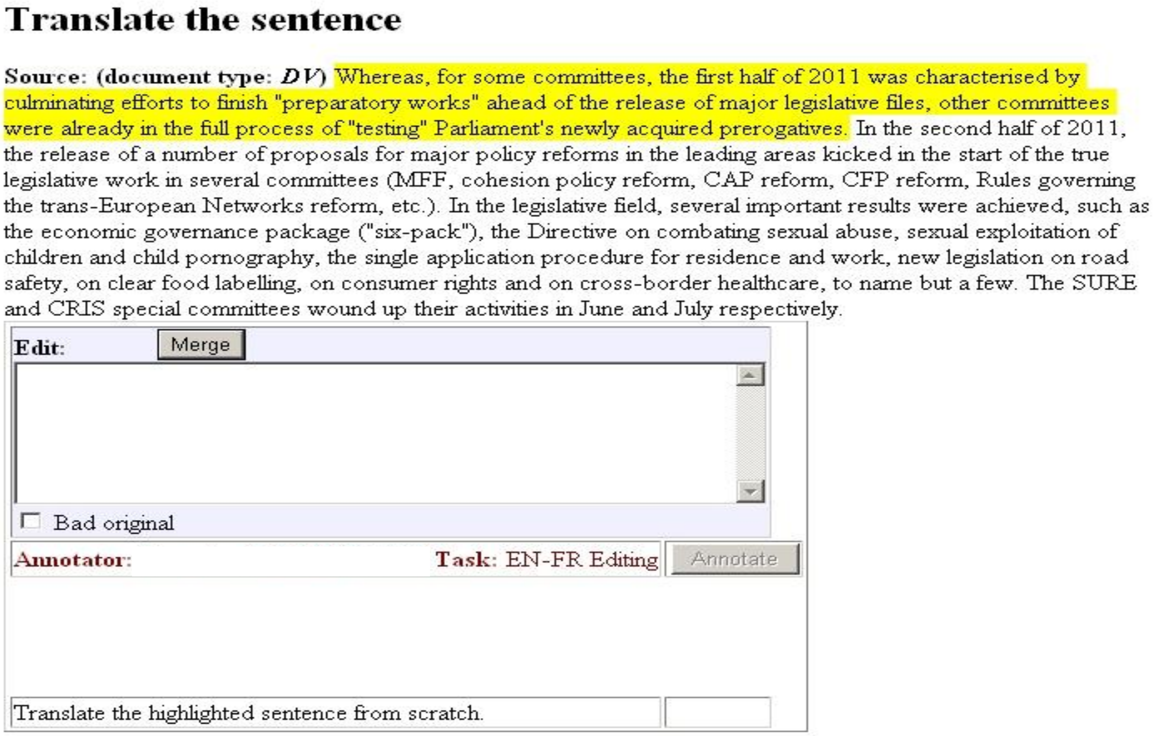
Un-edited MT Output:
Vu sa résolution du 30 novembre 2006 sur la mise en œuvre du programme communautaire de Lisbonne: Les petites et moyennes entreprises, notamment la politique en faveur des PME pour la croissance et l'emploi,

Edit:

Vu sa résolution du 30 novembre 2006 sur la mise en œuvre du programme communautaire de Lisbonne:

○ Perfect Translation, no editing needed
○ Edited
○ Not editable, translating from scratch would be faster
□ Bad original

Figure 3: Post-editing and translation task



Translate the sentence

Source: (document type: *DV*) Whereas, for some committees, the first half of 2011 was characterised by culminating efforts to finish "preparatory works" ahead of the release of major legislative files, other committees were already in the full process of "testing" Parliament's newly acquired prerogatives. In the second half of 2011, the release of a number of proposals for major policy reforms in the leading areas kicked in the start of the true legislative work in several committees (MFF, cohesion policy reform, CAP reform, CFP reform, Rules governing the trans-European Networks reform, etc.). In the legislative field, several important results were achieved, such as the economic governance package ("six-pack"), the Directive on combating sexual abuse, sexual exploitation of children and child pornography, the single application procedure for residence and work, new legislation on road safety, on clear food labelling, on consumer rights and on cross-border healthcare, to name but a few. The SURE and CRIS special committees wound up their activities in June and July respectively.

Edit:

□ Bad original

Annotator: **Task:** EN-FR Editing

Translate the highlighted sentence from scratch.

Figure 4: Translation task

Measuring post-editing time is certainly not enough to estimate the possible benefits of MT. 20% of the paragraphs displayed are not followed by MT output. The segments of these paragraphs have to be translated segment by segment from scratch to obtain reference values for each participant. Subsequently the translation throughput (words per hour) of one translator will be compared to her post-editing throughput. By "translation from scratch" we mean that no MT output is provided. Nevertheless, translators are able to use all the tools they usually have access to in their normal workflow. For obvious reasons the only resources they are not allowed to access are translation memories or documents that can provide them with complete translations of the segments displayed in the test application. To avoid possible bias towards post-editing or translation from scratch, in both cases translators are given access to the same tools and references. These tools are briefly presented in section 2.

4.2 What will be measured

The results of each test will be analysed separately for each language pair. The data collected from the categorization task will help us measure the quality of the tested MT solution at segment level. For this purpose the number of Excellent, Good, Medium and Poor segments will be reported whereby different segment lengths (short, medium and long) will be taken into consideration. To make sure that the results are consistent, intra- and inter-annotator agreement will be taken into consideration. This is possible thanks to the regular re-appearance of segments within the evaluation application. Intra-annotator agreement will be measured using the Kappa coefficient (Callison-Burch et al. 2012) and inter-annotator agreement will be estimated using the Fleiss kappa as presented in (Fleiss 1971).

In the post-editing task the time needed to post-edit a segment is the most important variable. This will be measured from the moment that a new segment is loaded until the last action on the page is taken. This action (editing or selection of radio button etc.) is not defined a priori because the test-users might select any sequence every time. Translation time is measured in the same way at the translation. Segments that appear in the post-editing task may not re-appear in the translation task. If a test-user encounters a sentence at the post-editing task and then is asked to translate it from scratch in the translation task there is no doubt that she will remember it and therefore translate it faster. Average post-editing and translation times per character may also be compared.

It is expected that users will adapt to the application as well as to the post-editing task itself. Therefore we also intend to measure individual change of post-editing speed over time taking into consideration each user's familiarity with the task.

To evaluate the current MT solution as a tool for gisting purposes we will compare the number of machine translated paragraphs that convey the meaning of the original text compared to those that do not.

4.3 Evaluation Data

Translation demand was the main criterion for the selection of the language pairs that are currently evaluated. As the vast majority of source documents are written in English test users were provided with data that have been machine translated from English to all official EU languages. The English translators have been provided with data translated from French to English and from German to English. French and German are the two other of the so called "pivot" languages. Although most translation units in the EP have translators that cover a very big number of languages (some of them master 6 languages or some times even more), there are certain language combinations that are very rare. For example when a document is drafted in Maltese and it has to be translated in Lithuanian it is not very likely to find an in-house translator who is able of translating between these two languages. The same is the case for other target languages of course. Therefore, many documents are translated into the three pivot languages first which are mastered by the majority of translators and subsequently into all official EU languages.

To gather a sufficient amount of data without increasing too much the translators' workload at the same time a total amount of 40 pages will be processed per language pair. Two or three translators have been made available for each language pair and they have two and a half months to accomplish the task.

4.4 Preliminary Results

At the time when this paper was written two translators had accomplished their categorization task and another 6 had reached at least 50%. The current results are summarized by language pair in Table 2.

Language Pair	Poor	Medium	Good	Excellent
EN-PL	25 %	30 %	34 %	11 %
EN-DA	4 %	17 %	51 %	29 %
FR-EN	34 %	14 %	16 %	36 %
EN-LT	50 %	30 %	10 %	10 %
DE-EN	48 %	13 %	17 %	21 %

Table 2: Preliminary results of segment categorization by language pair

With maximum two users for each language pair having completed in most cases roughly 60% of their categorization task these results can merely show a certain trend: at least 50% of all segments evaluated for each language pair are of medium quality and thus post-editable with this percentage reaching up to 96% for English to Danish. At this stage the used MT system seems to provide less usable results for EN-LT while according to direct feedback from the English evaluators DE-EN is rather problematic too with many results being of very poor quality. It should be added here that the two English evaluators that worked on DE-EN and FR-EN have accomplished their categorization task.

As expected these results are not consistent for all document types. For example 92% of segments coming from QO documents (oral questions) were judged as poor while other document types had much fewer or some times no segments at all judged as poor. Two possible reasons for the high number of poorly translated QO segments are data scarcity (not many QO documents in the training data) as well as the style and register used in these documents which is totally different from any other document type. So far most evaluators have shown a high intra-annotator agreement.

5 Conclusions

In this paper we described the evaluation methodology and some preliminary results of a state of the art statistical MT system at the European Parliament. With the use of post-editing and other MT evaluation tasks fine-tuned to our business needs we will use the collected data to estimate the benefits that DG TRAD may have from the implementation of MT technology in the current translation workflow as a complementary tool to Translation Memories, terminology databases, bilingual concordancers and other reference tools.

6 Future work

After the end of the current evaluation exercise we will try to use the collected data to estimate the expected business benefits.

The conclusions that will be drawn from this evaluation procedure will be used in the future as a baseline to avoid re-running similar exercises too often as they require the involvement of many human resources. Future evaluations will most probably ask the users to compare the output of the future MT engines to that of the current ones. A more detailed manual error-analysis will also be conducted to identify key areas of MT improvement. One such example could be specific grammar errors in morphologically reach languages which may be solved with language-specific rules.

The analysis of the annotation data will also help us understand our needs for post-editing training and come up with more precise specifications.

In the future we expect to integrate MT in the translation workflow in such a way that similar conclusions will be drawn in the real translation environment without creating extra work for translators. Creating this translation-feedback loop we expect to get more reliable results as our current method is similar but not identical to real translation conditions.

Acknowledgments

We would like to thank all members of the MT user group and principally Jouko Menger and Miguel Frank for their valuable help in the organisation of the tests as well as our business analyst, Joanna Lankamer for her support. We would also like to thank Andreas Eisele and his colleagues at the European Commission for their valuable help with data preparation.

References

Achim Blatt. 1998. EURAMIS: Added value by integration. In T&T Terminologie et Traduction, 1.1998, pages 59–73

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In Proceedings of the Fourth Workshop on Statistical Machine Translation, pages 1–28, Athens, Greece, March. Association for Computational Linguistics

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut and Lucia Specia. 2012. Findings of the 2012 Workshop on Statistical Machine Translation Proceedings of the Seventh Workshop on Statistical Machine Translation

Andreas Eisele and Caroline Lavecchia. 2011. Using statistical Machine Translation for Computer-Aided Translation at the European Commission. JEC 2011

Michael Jellinghaus, Alexandros Poulis and David Kolovratnik. 2010. Proceedings of the Fifth Workshop on Statistical Machine Translation

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. Psychological Bulletin, vol. 76, No 5, Pages 378-382

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. MT-Summit 2005

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In Proc. of ACL Demo and Poster Sessions, pages 177–180

Philipp Koehn and Jean Senellart 2010. Convergence of Translation Memory and Statistical Machine Translation. AMTA Workshop on MT Research and the Translation Industry

Johann Roturier. 2009. Deploying novel MT technology to raise the bar for quality: A review of key advantages and challenges. In The twelfth Machine Translation Summit, Ottawa, Canada, August. International Association for Machine Translation

Michel Simard & Pierre Isabelle. 2009. Phrase-based machine translation in a computer-assisted translation environment. MT Summit XII: proceedings of the twelfth Machine Translation Summit, August 26-30, 2009, Ottawa, Ontario, Canada; pp.120-127

D. Theologitis. 1997. EURAMIS, the platform of the EC translator. In EAMT Workshop, pages 17–32

How Good Is Crowd Post-Editing?
Its Potential and Limitations

Midori Tatsumi
Toyohashi University of Technology
midori.tatsumi2@mail.dcu.ie

Takako Aikawa
Microsoft Research, Machine Translation team
takakoa@microsoft.com

Kentaro Yamamoto
Toyohashi University of Technology
yamamoto@lang.cs.tut.ac.jp

Hitoshi Isahara
Toyohashi University of Technology
isahara@tut.jp

Abstract

This paper is a partial report of a research effort on evaluating the effect of crowd-sourced post-editing. We first discuss the emerging trend of crowd-sourced post-editing of machine translation output, along with its benefits and drawbacks. Second, we describe the pilot study we have conducted on a platform that facilitates crowd-sourced post-editing. Finally, we provide our plans for further studies to have more insight on how effective crowd-sourced post-editing is.

1 Introduction

As the use of machine translation (MT) together with post-editing (PE) has become one of the common practices to achieve cost-effective and high quality translation (Fiederer & O'Brien 2009, Koehn 2009), and crowdsourcing is gaining popularity in many areas including translation (Désilets 2010, Zaidan & Callison-Burch 2011), one can easily imagine that 'crowd PE' is going to be a strong trend in the MT community in the near future.

This paper presents a preliminary investigation on the effectiveness of crowd PE. We conducted a pilot study using Collaborative Translation Framework (CTF) developed by the Machine Translation team at Microsoft Research. Having CTF as a platform of crowd PE, we translated the English websites of Toyohashi University of Technology (TUT)[1] into nine languages with very little cost (Aikawa et al. 2012). We analysed the results from this pilot study quantitatively in an attempt to evaluate the validity and the effectiveness of crowd PE.

The organization of this paper is as follows: In section 2, we discuss the past and the current situation of crowdsourcing in text and contents production, and state the goal of our research. Section 3 presents a brief explanation of our pilot study at TUT and its results. In section 4, we provide some results from the human evaluation on the quality of the crowd PE, and the results from the evaluation by means of an automatic metrics. Section 5 discusses the results from Section 4, while raising our renewed research questions. Section 6 summarises the paper.

We are aware that building and maintaining appropriate platforms and communities is an important aspect of crowd PE, and a number of research efforts are being made on those topics. Our paper, however, is focused on the quality we can expect from crowd members, and thus building and maintaining platforms and communities is out of the scope of this paper.

[1] http://www.tut.ac.jp/english/introduction/

2 Crowd Post-Editing or 'CPE'

The power of crowd resource in producing translation has been proven in a number of areas from fansubs (Cintas & Sánchez 2006, O'Hagan 2009) to social media such as Facebook (Losse 2008) to popular conference video site, TED[2], to community participation in product development at Adobe[3] and Symantec (Rickard 2009). This makes one think: if crowd translation has been successful, why not crowd post-editing? It may not be too extravagant to even speculate that crowd PE has more potential than crowd translation; considering that crowd members are often not professional translators or linguists, PE may seem to them as a less demanding task than translating from scratch (though in reality PE of MT sometimes can be more demanding than translation depending on the MT quality).

In fact, researchers and businesses have already started to study and test the potential of this method (Muntes & Paladini 2012). However, the current focus is mainly on developing platforms to facilitate the participation of crowd members and frameworks for quality control. The actual quality of the crowd PE outcome has not yet gained much attention.

Crowd PE can have different types of resources. Some cases may hire random crowd resources with a small monetary reward (e.g., Amazon's Mechanical Turk), others may be done by enthusiastic fans of the subject matter, or some others may even employ only the internal members of an organisation or a community, involving no payment. The latter cases can be more appropriately called 'Community PE' or 'Collaborative PE' than 'Crowd PE'. In this paper, we do not differentiate these different types of resources, and will use the acronym 'CPE'.

2.1 Advantages of CPE

MT + CPE, similar to crowd translation, can be advantageous in a number of aspects compared to MT + professional PE (i.e., post-editing done by professional translators and post-editors). The following lists such advantages.

Cost: CPE is less expensive than professional PE, which is especially important for non-profit organisations and/or the types of contents that need to be updated frequently. This, however, only applies to the per-word cost, and the initial investment on developing the platform, framework, interface, etc. needs to be taken into account when evaluating the total cost.

Speed: Crowdsourcing often proves to be equally quick, or sometimes even quicker, than the traditional style commercial works[4].

Domain Knowledge: Although crowd members are not expected to have linguistic expertise, they are often highly knowledgeable in specific domains.

Community Development: Crowd members can get the sense of community by participating in CPE. In addition, CPE might give the contributors an opportunity to become more familiarised with the community topics and issues as they try to read and understand the contents more deeply than they would as a mere reader.

Confidentiality: CPE also has a potential to be an ideal solution for translating sensitive contents in an organisation. Translating the text by an MT system and have internal members to perform CPE can eliminate the fear for information leakage (provided enough resources can be secured within the organisation).

2.2 Drawbacks of CPE

One big challenge CPE would face is how to assure the quality of CPE. To address this issue, most, if not all, of the crowdsourcing platforms provide one or more ways to control the quality of the crowd-sourced products. One of the common methods is to have one or more moderators to check and ensure the quality of the product. Another common method is rewarding and/or ranking mechanism that gives various rewards and/or quality statuses to the crowd members based on the past performance. Such mechanisms are designed to encourage the participants to make more contribution with higher quality jobs.

[2] http://www.ted.com/OpenTranslationProject

[3] The blog article written by Dirk Meyer is available at: http://blogs.adobe.com/globalization/collaborative-translation-helps-adobe-business-catalyst-add-new-languages/

[4] One example is the translation of movie subtitles in China (Chipchase, J. & Wang, F. "subtitle team, crowd sourced translation in China". Available at: http://janchipchase.com/2011/09/chinese-bandit-translation-teams/).

These solutions can help to overcome the quality assurance issue, but it can also incur a great amount of effort and investment to develop and maintain complicated frameworks and platforms. If we know what level of quality we can expect from CPE, it would help to make a necessary and sufficient investment on quality assurance. This paper is a step stone to this goal.

3 Pilot Study

This section provides a brief description of our pilot project conducted at TUT, which we mentioned at the beginning of the paper.

3.1 Motivation and setting

TUT has more than 200 foreign students from various countries, and the demand to localise the information on their websites into various languages has always been strong. Yet, localising the websites using professional translators is just too expensive. To make the university information more accessible to current foreign students and to prospective students, the university created an English version of their websites. However, still many foreign students had problems in understanding the information because of the language barrier. To overcome this issue, the university decided to translate the English websites into nine languages by means of Microsoft Translator's Widget, and have their foreign students to post-edit the MT output.[5]

Foreign students at TUT were ideal crowd resource for this project as they are familiar with the contents of the TUT's websites, and they are willing to make a contribution to this project with a small monetary reward. We hired a total of 22 foreign students [6] with nine different language backgrounds shown in Table 1.

3.2 Conducting the CPE Session

Prior to starting the project, we gave the students a brief introduction on how to use CTF user interface and explained the background of the project. We also provided the following CPE guidelines:

Avoid over-editing: don't try to over-edit if the existing translation(s) (whether they are MT output or other human edits) are grammatical and readable.

Ignore stylistic differences: don't try to modify stylistic differences unless they are critical or matter for readability.

Start from scratch: if the quality of MT output is too low, provide your translation from scratch (as opposed to modifying MT output).

It is important to note here that we did not prevent the students from modifying existing CPE results provided by other students. The students are allowed to modify not only the MT output but also any one of the previous CPE results as they think is necessary.

We assigned each student 30 hours for performing CPE. The CPE sessions were conducted in November-December, 2011. The details on the workflow of the CPE and the design of CTF are provided in (Aikawa et al. 2012).

3.3 Results

Table 1 shows the descriptive statistics of the results of the pilot study.

Language	Participants	Sentences	Edits
Arabic	2	397	723
Chinese[7]	6	1637	2269
French	2	512	647
German	1	147	192
Indonesian	2	1285	1559
Korean	2	598	707
Portuguese	1	204	308
Spanish	4	1841	3643
Vietnamese	2	1341	1929

Table 1. Summary of the results

The Sentences column shows the number of the sentences that were edited, [8] and Edits column shows the total number of sentences resulted from

[5] This is a collaboration project between TUT and Microsoft Reseach. See Yamamoto et al. (2012) for our initial report.

[6] Strictly speaking, the total number of student participants was 21 as one of the students edited both Arabic and French MT output.

[7] This study involved only simplified Chinese.

[8] Note that there were cases where no CPE was provided as MT output were acceptable enough. We did not study such cases as the focus of this study is the effect of CPE, and not the quality of MT

CPE, for each language. The gap between the two indicates that some sentences have received multiple CPE. Following is an example where multiple CPE were performed for Spanish:

[English source text]

You must show this table to the banker before sending your money.

[MT output]

Se debe mostrar esta tabla para el banquero antes de enviar su dinero.

[First CPE result]

Se debe mostrar esta tabla al banquero antes de enviar su dinero.

[Second CPE result]

Se debe mostrar esta tabla al empleado del banco antes de enviar su dinero.

Overall, the figures in the table show that the combination of Microsoft Translator's Widget and CTF has been well adapted as a community translation environment such as university websites. We have received a fair number of CPE outputs from the participant students, which demonstrates their enthusiasm. Using the crowdsourcing power of the foreign students at TUT, the majority of the university's English websites was localised into nine languages within two months with inexpensive cost.

We asked the participant students to give feedback about their experience as a CPE contributor. The students, though not having professional translation experience or linguistic expertise, seemed to have worked quite comfortably and confidently in the provided CPE environment, and their overall feedback was very positive. They also mentioned that participating in this project as a CPE contributor gave them the strong sense of community.

Now the important question we need to ask is: how good was the quality of CPE? We address this question in the next section.

4 Quantitative Analysis

4.1 Human evaluation

Among the nine languages post-edited for this pilot study, we chose four languages that had higher number of sentences post-edited than other languages, namely, Chinese, Indonesian, Spanish, and Vietnamese, to evaluate the CPE results. To this end, we hired professional translators and asked them to choose the best translation among all the translations (which consist of MT output and CPE results) in the sense that it reflects the meaning of the source text. We advised them not to worry about stylistic or registry differences. We also asked them to provide their own translation in case none of the existing translations conveyed the correct meaning of the source text. To make this evaluation a blind test, we randomised the order of the MT output and all the CPE results. This way, the evaluators (professional translators) could not tell which translation was from MT or CPE based on the order of the sentences.

For the purpose of a cross-language comparison, we focused only on the test sentences that had been post-edited for all four languages; there were 567 such sentences.

The following table shows the frequency of the occurrences of single and multiple CPE for each of the 567 test sentences.

Number of CPE	Chinese	Indonesian	Spanish	Vietnamese
1	372	441	196	350
2	137	95	175	154
3	41	24	88	43
4	10	5	46	17
5	6	2	28	2
6	1		22	1
7			8	
8			1	
9			3	
Average Number of CPE	1.49	1.29	2.39	1.54

Table 2. Frequency of multiple CPE

According to Table 2, except for Spanish, more than 60% of the test sentences had only one CPE output, and more than 95% less than three CPE outputs.

Here we make a simple assumption: among all CPEs, the last one should be the best one, assuming that the last one is the result of the collective intelligence of all the CPE contributors worked on a given sentence. When a sentence is post-edited by more than one person, the second person onward can see not only the MT output but also the previous contributors' editing results, thus can gain better idea of what an acceptable translation should be like, by learning from other people's editing.

In order to find out if this is true, we distinguish the last CPE output from other CPE outputs. In the analyses and descriptions below, we will use the following terms:

MT: Machine Translation output

LCPE: The Last CPE output for each test sentence. When there is only one CPE output, it becomes the LCPE.

XthCPE: All CPE outputs other than LCPE.

Revision: Revised text provided by the professional translators.

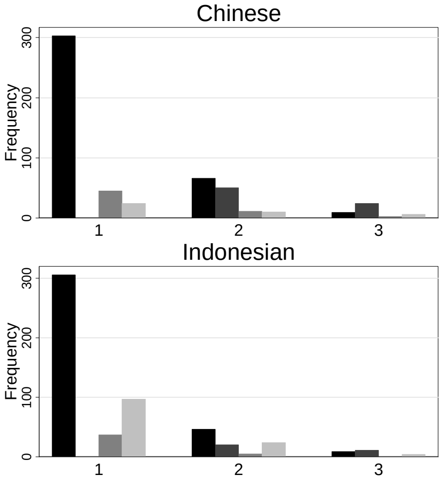
(When we just say ‘CPE’, it includes both XthCPE and LCPE.)

The following table shows the human evaluation results and the numbers of the cases where LCPE, XthCPE, or MT was selected or a Revision was provided for each language. The greyed area indicates the percentages. Note that when MT or XthCPE was selected and when it was exactly the same as LCPE, we counted that into LCPE. Likewise, when MT was selected and it was exactly the same as an XthCPE, we counted that into XthCPE.

Selected as Best/Revised	Chinese	Indonesian	Spanish	Vietnamese
LCPE	383	364	261	334
	68%	64%	46%	59%
XthCPE	85	34	154	67
	15%	6%	27%	12%
MT	58	42	50	22
	10%	7%	9%	4%
Revision	41	127	102	144
	7%	22%	18%	25%
Total	567	567	567	567

Table 3. Human evaluation results

Overall, LCPE is the most frequent choice for all languages, though the percentage varies from the highest of 68% for Chinese to the lowest of 46% for Spanish. This is generally good news, but it also means that our assumption that LCPE should be the best was not right for around 30 to 50% of the cases. XthCPE was selected as the best translation in 6 to 27% of the time, and MT 4 to 10% of the time. This means that one or more CPE contributors transformed the MT or existing CPE results that had acceptable quality into the one that did not. In order to further investigate this, we looked at the evaluation ratio by the number of CPE outputs. The following figures show the results (we only looked at the cases where one, two, or three CPE was performed, as there were not many cases for which more than three CPE outputs were available). Note that there is no bar for XthCPE for the category 1, as this is the case where there is only one CPE, that is, LCPE.



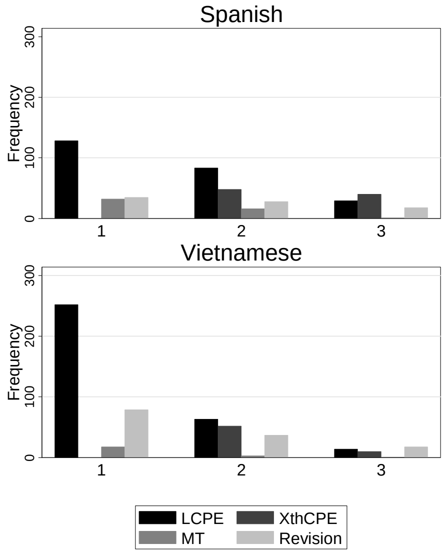


Figure 1. Relationship between the number of CPE output and the evaluation

As the figures show, for most of the cases where LCPE was selected, LCPE was the only CPE output (category 1). Interestingly, LCPE is still the best choice when one more CPE was done (category 2), but for the test sentences where CPE was performed three times (category 3), XthCPE was slightly more frequently chosen as the best translation, except for Vietnamese. This may mean that after the third CPE, the quality of the CPE output tended to deteriorate. We would like to investigate this issue further in the future.

There are 7 to 25% of the cases where professional translators did not find any satisfactory translation and provided a Revision. We were interested in finding out if there are any prominent source text characteristics that may have caused low quality CPE. As a starting point, we compared the average source sentence length in words between the sentences for which LCPE was chosen as the best translation and those for which Revision was provided. The following figure shows the result.

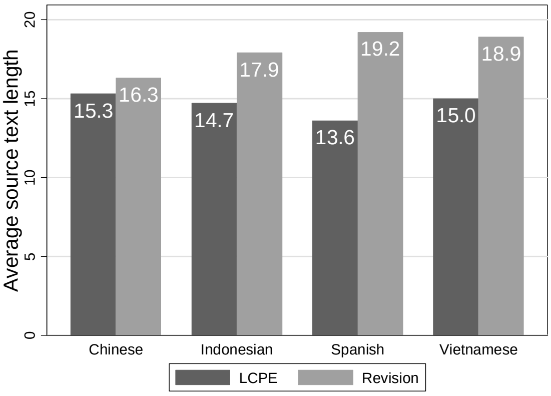


Figure 2. Comparison of source sentence length in two different cases

As shown in Figure 2, the average length of the source sentences that ended up having professional translators to provide the Revisions was longer than that of the sentences where LCPE achieved a good enough quality.[9] The average length for all 567 source sentences is 15.9 words.

4.2 Evaluation with TER

Next, we focused on two cases: Case I, where LCPE was selected as the best translation, and Case II, where the professional translator revised LCPE to produce acceptable translation.[10] This was to see 1) how much editing was done by CPE contributors in order to transform the MT output of unacceptable quality to the translation of acceptable quality, and 2) when LCPE was better than MT or XthCPE yet not quite good enough to be regarded as an acceptable translation, how much editing was necessary by the professional translators to produce Revisions.

To measure these, we used TER (Translation Edit/Error Rate)[11]. TER (Snover et al. 2006) is one of the automatic metrics developed for MT quality evaluation. It compares two sentences and calculates a score based on the number of minimum editing operations necessary to

[9] P<0.01 for Indonesian, Spanish, and Vietnamese. Statistical significance was not observed for Chinese.

[10] This, however, involves some subjectiveness. When the translator provided Revisions, the revised text is inserted next to the text that the translator thought was the closest to the acceptable translation. However, the revised text sometimes ends up in becoming closer to other text than the one they have chosen.

[11] http://www.cs.umd.edu/~snover/tercom/

transform one sentence to another. The perfect match gets a score of 0 (0 edits needed), and the score gets higher as the difference between the two sentences becomes larger. As it uses word as an editing unit, we used Stanford Chinese Word Segmenter[12] to tokenise the Chinese text.

We took TER scores between MT and LCPE for Case I, and between LCPE and Revision for Case II mentioned above. The average TER scores for the two cases are shown in Table 4.

Language	Case I: TER between MT and LCPE	Case II: TER between LCPE and Revision
Chinese	54	27
Indonesian	38	26
Spanish	40	34
Vietnamese	49	27

Table 4. Average TER for the two cases

The results show that, for Case I, Chinese got the highest score among four languages, which means that, on average, it took CPE contributors more editing to transform an MT output into an acceptable quality translation in Chinese than other languages.

On the other hand, for Case II, Spanish got the highest score, which means that it took professional translators more editing to fix LCPE to produce an acceptable level translation than other languages. We plan to investigate such language differences in more details in the future.

Overall, for all languages, the average TER scores between LCPE and Revision are significantly smaller than the scores between MT and LCPE.[13] This may suggest that even when LCPE could not achieve acceptable quality, the amount of Revision work necessary to improve such text to an acceptable level quality can be smaller than revising the MT output from scratch.

5 Discussions and Ongoing Studies

Overall, the above mentioned results suggest the following:

- Around 50 to 70% of the time LCPE produces good enough translation
- Longer source sentences may cause difficulty for CPE contributors to produce acceptable quality
- Even when LCPE result is not good enough, the amount of necessary additional revision work may be rather small

These results generally show that CPE can be a great help in raising the MT output quality to an acceptable level. However, there were still cases where professional translators found LCPE results unsatisfying or LCPE having lower quality than XthCPE or even MT. This gave us renewed research questions (RQ) listed below.

RQ1: In what kind of cases do CPE contributors fail to produce acceptable translations?

We found that the number of the cases professional found LCPE results unacceptable varies among the languages. However, the numbers alone do not tell us 'why'. In order to understand more deeply in what cases and in what way LCPE failed to produce an acceptable translation, we will need to examine the results qualitatively.

RQ2: Would having the larger number of CPE contributors be of help in achieving acceptable quality?

We found that 46 to 68% of the time LCPE was selected, but would the percentage increase if we ensure each MT output is post-edited by certain number of CPE contributors? Would the quality keep increasing to the point where the professionals' intervention becomes unnecessary?

In order to answer these questions, we are now in the process of the following two further studies.

5.1 Qualitative Analysis

In order to answer RQ1, we are having one native speaker of each target language, who has some translation experience, but not the same person who did the evaluation task explained in section 4.1, to explain the difference between CPE

[12] http://nlp.stanford.edu/software/segmenter.shtml

[13] $P<0.05$ for Spanish. $P<0.001$ for Chinese, Indonesian, and Vietnamese.

outcome and its Revision. The interview sessions will be held in August 2012.

Based on the results of the interviews, we are hoping to have insights into what kinds of necessary editing CPE contributors tend to achieve or fail to achieve, for each language, and also for all languages.

5.2 Controlled Experiment

In order to answer RQ2, we are conducting a controlled experiment in which all the sentences are ensured to be post-edited by certain number of CPE contributors.

We predict that, after certain number of editors, there will be nothing left to improve, and hence editing would become 'saturated'.

We are interested in finding out the following:

- *Would the percentage of LCPE selected by the professional translator increase when we have more CPE contributors?*
- *If that is the case, how many is enough?*

We are currently running an experiment to answer these questions.

6 Concluding Remarks

In this paper we first discussed the current situation and the potential of crowd PE. Then we explained our pilot study on the impact of crowd PE, presenting some quantitative results from the human evaluation and the evaluation by means of TER. Finally, we stated our further research questions and introduced our ongoing research effort.

Acknowledgments

This project was funded by International Affairs Division at Toyohashi University of Technology, and we would like to give special thanks to all the members of International Affairs Division for their support during the project. We are also thankful to Dr. Anthony Hartley for his support on conducting the experiment.

References

Aikawa, T., Yamamoto, K., & Isahara, H. (2012) The Impact of Crowdsourcing Post-editing with the Collaborative Translation Framework. In: Proceedings of the 8th International Conference on Natural Language Processing, Kanazawa, Japan

Cintas, J.D. & Sánchez P.M. (2006) Fansubs: Audiovisual Translation in an Amateur Environment. *The Journal of Specialised Translation,* 6, pp. 37-52

Désilets, A. (2010) Collaborative translation: technology, crowdsourcing, and the translator perspective. Introduction to workshop at *AMTA 2010: the Ninth conference of the Association for Machine Translation in the Americas*, Denver, Colorado, October 31, 2010; 2pp

Fiederer, R. & O’Brien, S. (2009) Quality and machine translation: a realistic objective? *Journal of Specialised Translation,* 11, pp. 52-74.

Koehn, P. (2009) A process study of computer-aided translation. *Machine Translation,* 23, 4, pp. 241-263.

Losse, K. (2008) “Achieving Quality in a Crowd-sourced Translation Environment”. Keynote Presentation at the 13th Localisation Research Conference Localisation4All, Marino Institute of Education, Dublin, 2-3 October 2008

Muntes, V. & Paladini, P. (2012) “Crowd Localization: bringing the crowd in the post-editing process”. Presentation at Translation in the 21st Century – Eight Things to Change, Paris, May 31 – June 1 2012

O’Hagan, M. (2009) Evolution of User-generated Translation: Fansubs, Translation Hacking and Crowdsourcing. *The Journal of Internationalisation and Localisation,* Volume I 2009, pp. 94-121

Rickard, J. (2009) Translation in the Community, Presentation at LRC XIV conference, Localisation in the Cloud, 24-25 September, Limerick, Ireland, available at http://www.localisation.ie/resources/conferences/2009/presentations/LRC_L10N_in_the_Cloud.pdf

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., & Makhoul, J. (2006) A Study of Translation Edit Rate with Targeted Human Annotation. In: Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, Cambridge, Massachusetts, USA, pp. 223-231.

Yamamoto, K., Aikawa, T. & Isahara, H. (2012) 機械翻訳出力の後編集の集合知による省力化. In: Proceedings of the 18th Annual Meeting of the Association for Natural Language Processing, Hiroshima, Japan, pp. 499-500

Zaidan, O. & Callison-Burch, C. 2011. Crowdsourcing translation: professional quality from non-professionals. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pp. 1220–1229.

Error Detection for Post-editing Rule-based Machine Translation

Justina Valotkaite
Research Group in Computational Linguistics
University of Wolverhampton
justina.valotkaite@gmail.com

Munshi Asadullah
Research Group in Computational Linguistics
University of Wolverhampton
asad.anto@gmail.com

Abstract

The increasing role of post-editing as a way of improving machine translation output and a faster alternative to translating from scratch has lately attracted researchers' attention and various attempts have been proposed to facilitate the task. We experiment with a method to provide support for the post-editing task through error detection. A deep linguistic error analysis was done of a sample of English sentences translated from Portuguese by two Rule-based Machine Translation systems. We designed a set of rules to deal with various systematic translation errors and implemented a subset of these rules covering the errors of tense and number. The evaluation of these rules showed a satisfactory performance. In addition, we performed an experiment with human translators which confirmed that highlighting translation errors during the post-editing can help the translators perform the post-editing task up to 12 seconds per error faster and improve their efficiency by minimizing the number of missed errors.

1. Introduction

Since its introduction Machine Translation (MT) has improved considerably and recently it has started gaining recognition in the translation industry. However, translations of MT systems have not yet reached the level of human quality. One of the ways of improving MT outputs is by performing the task of post-editing (PE), which nowadays, is becoming a common practice. According to Suzuki (2011), "to make the best of machine translation humans are urged to perform post-editing efficiently and effectively". As a starting point, in this study, we focus on Rule-based Machine Translation (RBMT), since we believe these systems produce errors in a more systematic manner, which makes capturing these errors more feasible.

In their outputs RBMT systems tend to repeat the same mistakes. Therefore, while post-editing, humans are forced to correct the same mistakes repeatedly and this makes the post-editing task draining and monotonous. In this study we aim at investigating a way of providing support for the post-editors by designing linguistically motivated rules for error detection that could be integrated into a post-editing tool. Our hypothesis is that these rules could help post-editors by indicating problems in the output which need to be fixed, and as a consequence help minimise post-editing time.

Recent work has addressed error detection and its visualization following shallow, statistic approaches for the error detection and focusing mostly on SMT. Koehn and Haddow (2009) introduced a tool for the assistance of human translators with functionalities such as prediction of sentence completion, options from the translation table and post-editing. Experiments with a Statistical Machine Translation (SMT) system and ten translators revealed that the translators were fastest when post-editing with the tool.

Xiong et al. (2010) proposed the integration of two groups of linguistic features, i.e. lexical and syntactic features, into error detection for Chinese–English SMT. These features were put together with word posterior probability features using a classifier to predict if a word was correct or incorrect. Various experiments were carried out and the results revealed that the integration of linguistic features was very useful for the process of error detection because the linguistic features outperformed word posterior probability in terms of confidence estimation in error detection.

Bach et al. (2011) proposed a framework to predict SMT errors at word and sentence levels for Arabic–English translation. They used a large dataset with words and phrases that had been previously post-edited as features for training the error detection model. As part of their experiments they also introduced a visualization prototype for errors in order to improve the productivity of post-editors by helping them quickly identify sentences that might have been translated incorrectly and need correction. Their method was based on confidence scores, i.e. predictions at phrase and word level for good, bad and medium quality translations. The results showed that the MT error prediction accuracy has increased from 69.1 to 72.2 in F-score.

While our goal is very similar to that of these papers, we address the error detection of the MT output from a more linguistically motivated perspective. We derive linguistic rules from an error analysis of Portuguese–English sentences from two text domains and two variants of Portuguese (European and Brazilian) translated by two RBMT systems – Systran[1] and PROMT.[2] We consider detection rules to be a practical and potentially helpful solution for RBMT systems, since these are known for making repetitive mistakes. In other words, if a system cannot cope with some specific language phenomenon, for example, recognizing a certain word or a construction in a source language (SL), it is likely to keep making the same mistake whenever that phenomenon is encountered. In addition to understanding whether we can successfully detect the errors with these rules and whether highlighting them can help human translators, we are interested in assessing how general the rules (and the errors made by the systems) are across MT systems of the same type (rule-based) and across significantly different text domains.

Un-translated words	Inserted article*
Inflectional error*	Incorrect preposition
Incorrect voice *	Inserted preposition*
Mistranslated pronoun*	Inserted pronoun*
Missing pronoun*	Incorrect adjective translation*
Incorrect subject-verb order*	Incorrect order of nouns and their adjectival modifiers*
Missing article*	Incorrect date translation format/ numbering system*
Incorrect other word order*	Incorrect article / an article replaced by another POS*
Incorrect lexical choices	Incorrect translation of Portuguese reflexive verbs
Repeated words	Incorrect translation of Portuguese weekdays
Added words	Translated Portuguese surnames*
Missing words	Translated Portuguese abbreviation*
Main message is different	Missing subjects/ predicates
Capitalization problems*	POS error: a verb instead of an adjective etc (the same root)*
Missing if-clause*	Missing preposition*

Table 1. Error classification for the English-Portuguese language pair. Categories marked with * were later modelled by rules (Section 3)

In the remainder of this paper, we first describe our linguistic analysis (Section 2), to then describe the implementation of the rules (Section 3**)** and present a post-editing experiment with human translators (Section 4).

2. Linguistic Analysis

For the linguistic analysis we randomly selected 300 Portuguese sentences from two corpora: 150 sentences from Europarl (Koehn 2005), which is a collection of parliamentary speeches representing European Portuguese, and 150 from Fapesp (Aziz and Specia 2011), which is a collection of scientific news for Brazilian Portuguese. The minimum length of Europarl sentences was 3 words, maximum – 115, average – 27; whereas the minimum length of Fapesp sentences was 3 words, maximum – 88, and average 31. We then translated these sentences into English using Systran and PROMT, totalling 8,455 words.

The linguistic analysis was carried out as follows. First, we manually analyzed each sentence, identified various translation errors and

[1] http://www.systran.co.uk/

[2] http://www.promt.com/

assigned them to different error categories. We identified errors by correcting the sentences until they were of acceptable quality but at the same time trying to keep them as close as possible to their machine translated versions. In cases when errors co-occurred, i.e. two categories could have been applied for the same issue, these were counted twice. The error classification introduced in this paper was inspired by the classification schemes introduced by Flanagan (1994), Farrús (2010), and Specia et al. (2011).

Table 1 presents the error classification. The current analysis showed that the Portuguese–English translation outputs contained the most frequent and typical language-independent MT errors, such as "Incorrect lexical choice", "Inflectional errors", "Untranslated words". On the other hand, we also identified some language-specific errors, typical to the Portuguese-English language pair. Table 2 illustrates a subset of these: the most frequent error categories identified during the linguistic analysis.

Error category	Percentage of total errors			
	Systran		PROMT	
	Europarl	Fapesp	Europarl	Fapesp
Incorrect lexical choices	31.29	31.73	34.41	34.56
Inflectional error	9.84	5.61	7.76	5.87
Mistranslated pronoun	9.52	6.41	9.22	5.87
Untranslated words	4.19	4.33	2.20	6.21
Incorrect other word order	8.39	5.93	4.83	3.02

Table 2. The most frequent error categories in the corpora and systems analysed

An interesting example of language-specific category is the "Incorrect translation of Portuguese weekdays". In Portuguese, names of weekdays except *sábado* (*Saturday*) and *domingo* (*Sunday*) are compounds made of two individual words: a numeral and a noun. For instance, *segunda-feira* (*Monday*), *quinta-feira* (*Thursday*), etc. However, both systems failed to produce correct translations for these compounds. Instead, Systran produced a literal translation for both individual words (*quinta-feira* was translated as *fifth-fair*); whereas, PROMT translated them as equivalent weekdays in English but also added a verb phrase which was not present in the source text and did not make sense in the given context (*quinta-feira* was translated as *Thursday-sells at a fair*).

Based on the frequency of the errors in each category and on an analysis on the feasibility of creating rules for them, we selected 20 categories (marked * in Table 1) for which we designed the rules that later could be implemented and used to support the post-editing task.

We created the rules by analysing the errors of the target language (TL) sentences and comparing these to their corresponding SL sentences. In total, we produced a set of 40 contrastive rules which covered various problematic linguistic issues. For instance, if the systems made mistakes while dealing with present, past and future verb tenses and chose incorrect tenses for the TL translations, rules were designed for these specific issues. It is important to emphasize that although the two variants of Portuguese are considerably different, we focused on creating the rules that apply for both.

Rules are of the if-then type: *if in Portuguese <...>, then in English* <...>. For example, "If the verb X is in the past simple tense in Portuguese, then in English the translation of the same verb X must be in the past simple tense".

We then selected a subset of these rules which could be implemented for a pilot study. We took the following rules dealing with tense and number errors for the inflectional category due to the availability of the necessary pre-processing tools for the two languages and because they represented one of the most frequent error types in the output sentences:

- If the noun X is in singular/plural in Portuguese, the translation of the noun X should also be in singular/plural in English.
- If the verb X is in the 1st/2nd/3rd person singular/plural in Portuguese, the translation of the verb X should also be in the 1st/2nd/3rd person singular/plural in English;

- If the verb X is in the infinitive/ present simple tense/past simple tense/future simple tense in Portuguese, the translation of X in English should also be in the infinitive/ present simple tense/past simple tense/future simple tense;
- If the Portuguese construction contains “ir (*to go*) + infinitive”, then the English translation should be the future simple tense “(subjective pronoun) + will + simple verb/(subjective pronoun) / to be + going to + infinitive” (e.g. vou falar = I will speak; vamos verificar = we are going to check);
- If the Portuguese verb construction contains “não + V_{prs} + 3^{rd} person sg”, the English equivalent should be the construction “(subjective pronoun) + auxiliary verb + 3^{rd} p. sg. + not + infinitive” (e.g. não fala = (subj. pronoun) does not speak);
- If a Portuguese verb phrase is of progressive aspect, i.e. “estar + a + infinitive”, it should be in the present continuous tense in English “subjective pronoun + the form of to be + V_{ing}” (e.g. está a falar = (s)he is speaking).
- If the Portuguese verb construction is the following “Vprs +3^{rd} p. sg. + a + infinitive, in English it should be the English construction V_{prs} +3^{rd} p. sg. + infinitive” (e.g. ajuda a conter = helps to contain).

3. Evaluation of the Categories and Rules

In order to analyse how systematic the selected categories are and check the coverage of the rules created and the ones selected for the implementation, we performed two small scale experiments on two new datasets. For the first experiment we randomly selected 100 additional sentences from the original corpora: 50 from Europarl and 50 from Fapesp, and translated them using Systran and PROMT. The output of both datasets resulted in 5,676 words. The minimum length of Europarl sentences was 3 words, maximum – 68, average – 26; whereas the minimum length of Fapesp sentences was 3 words, maximum – 62, average – 30.

During the analysis, we fixed the translations to identify translation errors as before and assigned them to our error categories. The results revealed that out of our 30 categories, 26 were present in the new dataset, despite its smaller size. Only four categories - “Missing if-clause”, “Incorrect adjective translation”, “Missing subjects/predicates” and “Incorrect translation of Portuguese weekdays” - were not found in the new dataset. It is also important to emphasize that no new categories were identified in this dataset. From these results it can be concluded that the error classification for the Portuguese–English language pair in Table 1 is representative of these two text domains and RBMT systems.

Furthermore, we checked the frequency of all errors and in particular those of the inflectional category. The translations in the new dataset contained 393 errors, out of which 54 were attributed to the inflectional error category. To verify the coverage of the rules and in particular of those dealing with inflectional errors, we computed the percentage of errors in the new dataset that could be dealt with by the rules derived for the original dataset. The coverage was computed by dividing the number of errors for which there were no rules created by the total number of errors. The results showed that the coverage of the whole set of rules was 98.21%, while the coverage of the rules of the inflectional category was 92.59%.

For the second evaluation experiment we randomly selected 100 sentences from two new corpora: 50 sentences from CETEMPublico[3] which covers news in European Portuguese, and 50 sentences from CETENFolha[4] which covers news in Brazilian Portuguese. The minimum length of CETEMPublico sentences was 6 words, maximum – 58, average – 27; whereas the minimum length of CETENFolha sentences was 11 words, maximum – 51, average – 24.

We translated the sentences using both RBMT systems, resulting in 5,039 words. Once again, we checked the coverage of the categories and rules. The results revealed that all 30 categories introduced in the error classification were present in this dataset and no new categories were identified. In total, the output sentences contained 513 errors, with 39 of them of the inflectional

[3] http://www.linguateca.pt/cetempublico/informacoes.html
[4] http://www.linguateca.pt/cetenfolha/

category. The coverage of the whole set of rules was 93.67%, while the coverage of the inflectional rules selected for implementation was 87.18%.

From these results we can conclude that it is possible to systematically categorise errors in RBMT systems and that linguistic rules with sufficient coverage can be created in new datasets for such categories.

	Europarl-Systran	**Europarl-PROMT**	**Fapesp-Systran**	**Fapesp-PROMT**
Recall	0.46	0.70	0.66	0.63
Precision	0.62	0.58	0.42	0.37
F-Measure	0.53	0.63	0.51	0.47

Table 3. The evaluation of the system

4. Implementation and evaluation of rules

A few pre-processing tasks were performed in order to obtain certain linguistic information necessary for the implementation of the rules, such as a part-of-speech, lemma, morphological information (number and gender). First we performed word alignment between the Portuguese-English sentence pairs by using the aligner GIZA++ (Och and Ney, 2003). GIZA++ aligns sentences at the token level producing in this case four pairs of the alignment datasets. As some incorrect alignments were found in the sentences, we manually corrected them in order to obtain a "clean" dataset, that is, a dataset that allows evaluating the rules themselves, isolating any effect of low quality word alignments. Each aligned sentence pair was checked and the necessary corrections were performed, i.e. some incorrect aligned links were deleted, while other necessary links were inserted.

The sentences were also parsed in order to obtain their morphological information. For this purpose we used the parsers Palavras [5] for Portuguese and ENGCG [6] for English, both available online. The Palavras parser was reported to have 99.2% correct morphological tagging (Bick, 2000) and ENGCC was reported 99.8% recall in morphological tagging (Voutilainen and Heikkilä 1994). Any other parser producing morphological information could in principle be used. For this pilot study, seven rules dealing with errors of tense and number agreement were implemented using Python.

We then analysed the performance of the rules manually, i.e. each output sentence was checked individually to find out how many translation errors the system identified correctly, how many were missed and if it identified any false positives. Precision, Recall and F-measure were calculated. The results are shown in Table 3.

The system found a high number of false positives in the output sentences. The main reasons for this is parser errors and inconsistencies and the fact that often more than one rule can be applied and no rule precedence scheme was defined at first. An example of a case where multiple rules could be applied is the following Portuguese construction "*the present tense form of ir* (*to go*) + *infinitive*". It expresses a future action and usually is translated into English using the future simple tense i.e. *vou / vais / vai / vamos / vão + falar = I / you /(s)he / we / they will speak*. However, when the system encountered this construction, it identified the correct English translation as incorrect. For example:

(1-PT): *Vou verificar* se nada de isso foi efectivamente feito.
(1-EN): *I will check if nothing of that was effectively an act.*

This happened because the first rule processed by the system (by default) stated that "*if the verb X is in the present simple tense, the English translation of the same verb X should be in the present simple too*". Therefore, when the system encountered *vou verificar*, it did not recognize the pattern of *vou + verificar* as a future tense construction but rather only the verb in the present simple tense *vou* and in the English side it expected to find *I go + check.* However, further in the list there was a rule explaining this specific pattern and indicating the correct translation. Defining rule precedence schemes is not a trivial problem. While this was possible for our small set of rules, this issue will require further investigation as this set grows to incorporate other linguistic phenomena. .

[5] http://beta.visl.sdu.dk/constraint_grammar.html
[6] http://www2.lingsoft.fi/cgi-bin/engcg

The largest number of false positives occurred due to the flaws of the parsers. Example (2) illustrates a false positive case due to errors of the parser. The system flags the English verb *comments* (3^{rd} person singular) as an error although it was correctly translated. This happened because the parser identified *comments* as a plural noun, and not as a verb. Therefore, when the system found *comenta (comments)* as a verb in present tense (3^{rd} person singular), it expected to find a verb in the English side.

(3-PT): *"Queremos aumentar o intercâmbio com instituições internacionais que são referência em a pesquisa em música e ciência", comenta Ferraz.*

(3-EN): *"We want to increase the exchange with international institutions that are a reference in the inquiry in music and science", comments Ferraz.*

Some examples when the rules correctly detect translation errors include the incorrect tense and number translation (4) and the incorrect translation (5), i.e. a noun instead of a verb:

(4-PT): *Todos os restantes discordavam*.

(4-EN): *All the remainder was disagreeing* (*disagreed*).

But in that moment we were covering (we covered) almost only projects with support of the Fapesp, which was not the case.

(5-PT): *Penso que porque, mesmo mantendo as posições marxistas dialéticas, o ensaio era uma desmontagem de o marxismo fechado.*

(5-EN): *Bandage (I think) that because, even maintaining the dialectic Marxist positions, the test was a desmontagem of the shut Marxism.*

The implementation and the evaluation of the system showed that it is possible to have a working rule-based system which can detect certain translation errors using linguistic rules. Although in the current version some errors still remain to be captured due to their complexity and the limitations of the approach (the rules cover a small range of translation problems and only a sentence boundaries), we believe that error detection based on linguistic information is a promising direction to improve MT quality. While having a large number of false positives can still be an issue, this is less problematic than missing true errors. Although further experiments are necessary in that direction, our preliminary analysis in Section 6 indicates that translators can miss certain errors if these are not highlighted.

5. Experiments with Human Translators

A post-editing experiment was carried out with human translators in order to determine the usefulness of having errors highlighted in the RBMT output. Here we aim to investigate whether it is possible to help the human translators perform the task of post-editing faster and more efficiently when the MT errors are detected and highlighted.

To proceed with the experiment, we used the post-editing system PET (Aziz et al., 2012).[7] The tool gathers various useful effort indicators while post-editing is performed. We measured the *time* translators spent post-editing sentences. The tool also renders HTML, so highlighting errors was trivial.

For the test set in this experiment we randomly selected 60 sentence pairs from both Europarl and Fapesp. These sentences were then manually annotated, i.e. translation errors in the English as well as their corresponding source segments in the Portuguese sentences were marked. We resorted to manual highlighting rather using the errors detected by our system due to its limited coverage (only certain inflectional errors) and its relatively low performance.

After the manual error annotation, the sentence pairs were given to six human translators. We divided the test set into two parts, i.e. 30 sentence pairs with no errors highlighted and 30 sentence pairs with errors highlighted. All translators post-edited sentences with and without highlights. As the sentences were randomly selected, they contained different numbers of mistakes. Therefore, we analysed time on a per error (and not per sentence) basis. The errors were highlighted using different colours, each colour representing an individual error type from the set of 20 categories.

We produced guidelines in order to help human translators perform their task by explaining in detail how to use the post-editing tool and how they were expected to perform the task. All

[7] http://pers-www.wlv.ac.uk/~in1676/pet/

participants were native speakers of Portuguese. They were fluent in English and had some experience with translation tasks. European Portuguese translators were given European Portuguese sentences, whereas Brazilian translators post-edited Brazilian Portuguese sentences. We asked translators to post-edit machine translated sentences by making as few changes as possible and in such a way that the sentences would be grammatically correct and understandable. For sentences with errors highlighted, translators were also asked to evaluate the usefulness of having errors highlighted by choosing one of three possible options:

- Very useful.
- Some of them were useful.
- Not useful at all.

Translators were given one week to perform the task. Results were computed for three main aspects: the number of correctly identified and missed errors, the time taken to post-edit sentences in both datasets and the translators' evaluation of the usefulness of the highlights.

We manually analysed each translator's work by comparing their post-edited sentences with the previously annotated sentences. The results revealed that in the test set of the non-highlighted (**NH**) sentences the range of correctly identified errors by translators varies from 90% to 95.56%. The results for the sentences with the highlights (**WH**) showed a noticeable improvement, i.e. from 95.24% to 100%. It can thus be concluded that the performance of the translators improved when post-editing sentences with errors highlighted as the number of errors missed was significantly reduced. The reasons for translators missing errors in the non-highlighted sentences could be various, including the fact that perhaps the translators got used to having errors highlighted and the fact that some errors were not very significant for adequacy purposes, for example an incorrect extra article. However, the experiment showed that highlighting errors can be very helpful in attracting the attention of translators.

In order to find out if there was any significant difference between performing the two tasks in terms of time, we counted each translator's average time per error for both datasets. To get these estimates we divided the total time spent for post-editing each dataset (WH and NH) for each translator by the total number of errors in that dataset. The results are shown in Figure 1.

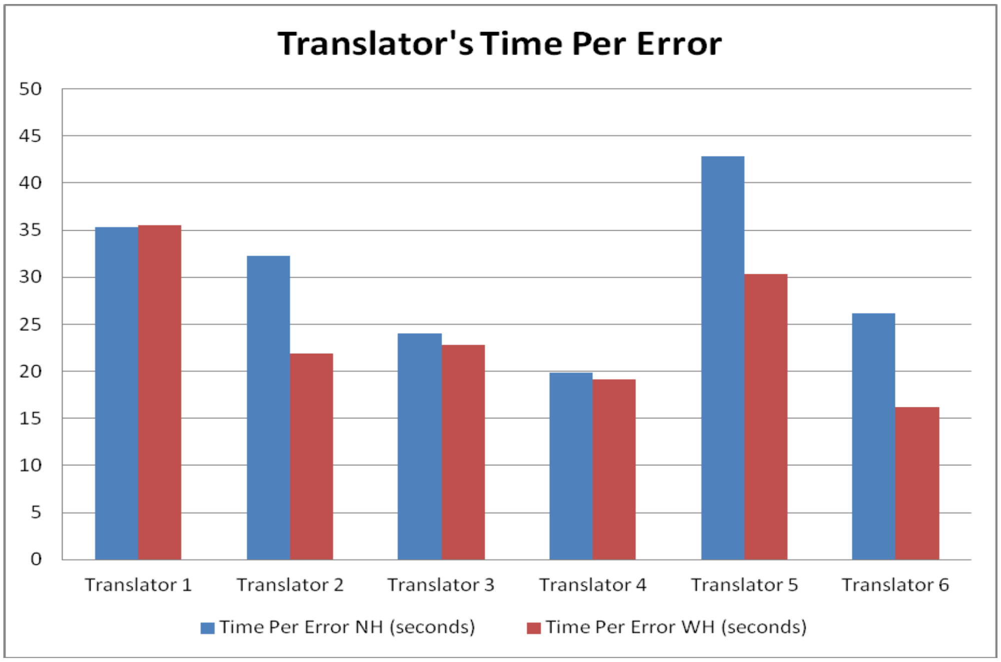


Figure 1. Time per error taken to post-edit the sentences

As it can be seen from Figure 1, translators' time per error NH and WH varies from 33 to 23 seconds (Translator 2), from 43 to 31 seconds (Translator 5) and from 27 to 17 seconds (Translator 6). These translators post-edited the WH sentences 10-12 seconds faster than the NH sentences.

For the rest of the translators, the results were less significant. The time of Translators 3 and 4 for the WH sentences was slightly better than for the NH sentences, varying from 22 to 24 seconds, and from 17 to 19 seconds respectively. On the other hand, Translator's 1 time was 36 seconds for the NH sentences and 35 seconds for the WH sentences. This could be explained by the fact that these translators missed a considerable number of errors, thus it is not surprising that there was no improvement in their results in terms of time. Although these improvements seem to be modest, we believe that when one extrapolates them to thousands of sentences with potentially dozens of errors, having errors highlighted can make a considerable difference in productivity.

The final factor which we analysed in this experiment was the opinion of the translators about the usefulness of the highlighted errors. As mentioned before, after post-editing each sentence with highlights, translators were asked how useful the highlights were. The results show that in 68% of the cases the highlights were found to be very useful, in 27% of the cases - some of them were found to be useful, and in only 5% they were found not to be useful at all.

The results of the experiments with human translators showed that the having errors highlighted can help human translators perform the task of post-editing faster and more efficiently. The highlights were also positively evaluated by translators.

6. Conclusions and Future Work

We have shown that one can systematically categorize translation errors which RBMT systems make and create linguistic rules for the error categories identified. We also showed that the rules apply across MT systems and text domains, and that one can implement a system detecting certain translation errors on the basis of those rules. Having a linguistically motivated approach for the error detection has also been shown to be helpful for the post-editing task. The results of a post-editing experiment with human translators revealed that the highlighted errors in the RBMT output helped to perform the PE task faster up to 10-12 seconds per error and improve translators' efficiency in identifying errors by reducing the number of errors missed. In addition, the highlighted errors were positively evaluated by the translators. Thus it can be concluded that the approach for post-editing based on the error analysis and the automatic error detection is promising and should be elaborated further.

The major challenge for future work is to scale up the approach. In order to implement the remaining rules, more levels of linguistic pre-processing will be necessary, such as named entity recognition. More robust ways of dealing with flaws in linguistic processors (such as the current parser issues) are also necessary.

Acknowledgment

This project was supported by the European Commission, Education & Training, Erasmus Mundus: EMMC 2008-0083, Erasmus Mundus Masters in NLP & HLT program.

References

Aziz, W. and Specia, L. (2011) *Fully automatic compilation of Portuguese-English and Portuguese-Spanish parallel corpora*. In: 8th Brazilian Symposium in Information and Human Language Technology (STIL-2011), Cuiaba, Brazil

Aziz, W., Sousa, S. And Specia, L (2012*) PET: a tool for post-editing and assessing machine translation.* In: The Eighth International Conference on Language Resources and Evaluation, LREC '12, Instambul, Turkey

Bach, N., Huang, F., and Al-Onaizan, Y. (2011). *Goodness: A method for measuring machine translation confidence*. In: 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, 211–219.

Bick, E. (2000). *The parsing system "Palavras" - automatic grammatical analysis of Portuguese in a constraint grammar framework*. In: Proceedings of the 2nd International Conference on Language Resources and Evaluation, TELRI, Athens

Farrús, M. Costa-Jussà, M., Mariño, J., and Fonollosa, J. (2010) *Linguistic-based evaluation criteria to identify statistical machine translation errors*. In: Proceedings of European Association for Machine Translation (EAMT), Saint Raphael, France, 52–57.

Flanagan, M. (1994) *Error classification for MT evaluation*. In: Technology Partnerships for Crossing the Language Barrier. Proceedings of the First Conference of the Association for Machine Translation in the Americas. Columbia, Maryland, US, 65-72.

Koehn, P. (2005) *Europarl: a parallel corpus for statistical machine translation*. In: Proceedings of the 10th Machine Translation Summit, AAMT, Phuket, Thailand

Koehn, P. and Haddow, B. (2009) *Interactive Assistance to Human Translators using Statistical*

Machine Translation Methods. In: MT Summit XII, 73-80

Och, F. and Ney, H. (2003) *A Systematic Comparison of Various Statistical Alignment Models*. Computational Linguistics, vol. 29, no 1, 19-51

Suzuki, H. (2011) *Automatic Post-Editing based on SMT and its selective application by Sentence-Level Automatic Quality Evaluation*. In: Thirteenth Machine Translation Summit (AAMT), 2011, Xiamen, China, 156-163.

Voutilainen, A. and Heikkilä, J. (1994) An English constraint grammar (EngCG): a surface-syntactic parser of English. In Fries, Tottie and Schneider (eds.), Creating and using English language corpora, Rodopi, 189-199

Xiong, D., Zhang, M., Li, H. (2010) *Error detection for statistical machine translation using linguistic features.* In: ACL: the 48th Annual Meeting of the Association for Computational Linguistics, *Uppsala, Sweden,* 604-611.

Machine Translation Infrastructure and Post-editing Performance at Autodesk

Ventsislav Zhechev
Autodesk Development Sàrl
Rue de Puits-Godet 6
2000 Neuchâtel, Switzerland
ventsislav.zhechev@autodesk.com

Abstract

In this paper, we present the Moses-based infrastructure we developed and use as a productivity tool for the localisation of software documentation and user interface (UI) strings at Autodesk into twelve languages. We describe the adjustments we have made to the machine translation (MT) training workflow to suit our needs and environment, our server environment and the MT Info Service that handles all translation requests and allows the integration of MT in our various localisation systems. We also present the results of our latest post-editing productivity test, where we measured the productivity gain for translators post-editing MT output versus translating from scratch. Our analysis of the data indicates the presence of a strong correlation between the amount of editing applied to the raw MT output by the translators and their productivity gain. In addition, within the last calendar year our system has processed over thirteen million tokens of documentation content of which we have a record of the performed post-editing. This has allowed us to evaluate the performance of our MT engines for the different languages across our product portfolio, as well as spotlight potential issues with MT in the localisation process.

1 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 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.

In this paper, we focus on the effect of the integration of MT in our localisation workflows. We start in Section 2 with an in-depth look at our MT infrastructure. Section 3 focuses on the productivity test we organised to evaluate the potential benefit of our MT engines to translators. In Section 4, we turn to the analysis of our production post-editing data from the last calendar twelve months. Finally, we conclude in Section 5.

2 MT Infrastructure at Autodesk

In this section, we present the MT infrastructure that we have built to support the localisation effort at Autodesk. We actively employ MT as a productivity tool and we are constantly improving our toolkit to widen our language coverage and achieve better perceived 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 are introducing MT for translating into Hungarian as a pilot this year.

2.1 MT Training Workflow

We start with the training of our MT engines.

Training Data

Of course, no training is possible unless sufficient amount of high-quality parallel data is available. In our case, we create the parallel corpora for training by aggregating data from three internal sources. The smallest source by far consists of translation memories (TMs) used for the localisation of marketing materials. The next source are our repositories for translated 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 are 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 based on product or domain. Instead, we lump all available data for each language together and use them as one single corpus per language. The sizes of the corpora are shown on Chart 1.

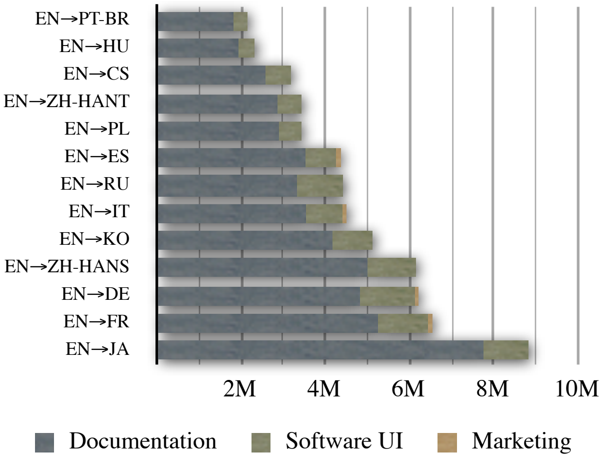


Chart 1: Training Corpora Sizes in Millions of Segments

You may notice that we have the least amount of data for PT-BR and HU, while our biggest corpus by far is for JA. You can refer to this chart when we discuss the evaluation of MT performance — it turns out that language difficulty is a stronger factor there than training data volume.

Data Preprocessing

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. The first step in this training workflow is the preprocessing of the data, which we turn to now.

For the majority of the languages that we support, the preprocessing step consists simply of tokenisation, masking of problematic characters and lowercasing. Some languages require additional preprocessing and we will discuss the details later in this section.

To perform the tokenisation, we have developed a custom Perl tool that consists mostly of a cascade of highly specialised regular expressions. We opted for this tailored approach as our data contains a large number of file paths and URLs, as well as specific formatting conventions and non-content placeholders that could be broken by a non-specialised tool. We also built abbreviation lists based on abbreviations observed in our data.

Another preprocessing step is lowercasing, which is a standard procedure used to improve lexical coverage and reduce lexical ambiguity.

The preprocessing scripts are used both to prepare the corpora for the MT engine training and to handle any data that has been received for translation at run time.

Data Post-processing

Although this is not a part of the training workflow, we will have a quick look at the post-processing tools we use, as they are closely related to the preprocessing tools we just discussed.

Post-processing takes place after a sentence has been translated and we have obtained the translation from the MT engine. As we tokenise and lowercase the data before training, we need to restore the proper case and detokenise the output of the MT engine to make it usable to humans.

For the former task, we use a statistical recaser. This is realised as an additional monolingual MT engine per language which is trained to translate lowercased input into proper-case output. Of course, this adds an additional element of uncertainty and opportunity to produce errors, but with the amount of data that we have available the performance is subjectively reasonable. On the other hand, it is much simpler to maintain statistical recasers — they are trained each time we train the regular MT engines — rather than rule-based recaser tools. The latter might require constant adaptation as new data is added to our TMs.

In an effort to recover from some potential errors the statistical recaser might introduce, we have added two specific rules. The first makes sure that the sentence-initial capitalisation of the MT output matches that of the English input. The second rule handles the capitalisation of unknown tokens. These tokens will most likely be new variable names or new URLs that the MT engine does not recognise. The recaser is not able to restore the proper case, which leads to hard-to-detect errors and frustration for the translators. Thus, we make sure that the casing of unknown tokens in the final MT output matches the provided input.

The detokenisation is a much simpler task and is realised as a cascade of regular expressions.

Language-specific Processing

Due to their specific makeup, some languages require extra preprocessing before we are able to handle them with MT. In our case, these languages are JA, KO, ZH-HANS and ZH-HANT.

Firstly, JA, ZH-HANS and ZH-HANT do not use spaces in written text, which makes it impossible to directly use a phrase-based MT system like Moses. We need to segment the data into word-like tokens that will then be used to align against English words. From the available tools on the market, we chose the open-source tool KyTea (Neubig et al., 2011), because it allows us to handle all three languages in question with the same process.

As expected, after translation we need to reverse these preprocessing actions to produce the final MT output. The de-segmentation for ZH-HANS and ZH-HANT is straightforward. We need to take extra care when desegmenting JA, however, as there are cases where the spaces need to remain in place — mostly within transliterated multipart English terms.

A harder issue to resolve with JA arises from the significant difference in syntactic structure between EN and JA, namely, EN is a Subject-Verb-Object language, while JA is a Subject-Object-Verb language. Hence, the linear distance between the verb in the EN source and its translation in the JA target may be very big making it difficult to handle by a phrase-based system like Moses.

Our solution to the problem is to reorder the EN source to make it more Japanese-like, thus reducing the linear distance between corresponding tokens in the EN and JA sentences. First, the EN source is assigned its phrase-based syntactic structure using the OpenNLP parser (`opennlp.apache.org`). Then, we use a rule-based tool developed in-house to move the syntactic heads of the EN sentence to positions corresponding to JA syntax. Our tests have shown this reordering to significantly increase the translators' post-editing productivity, compared to translating from scratch. In fact, using a plain (non-reordered) JA engine does not lead to a meaningful productivity increase, even though we have by far the largest amount of parallel data for the pair EN→JA compared to our other corpora.

Improvements to the Moses Training Toolkit

As stated above, we use the *de facto* standard Moses toolkit for training and decoding. However, early in the process of integrating MT in our localisation workflow, we ran into resource issues during the MT training. The main problem for us was that we could not reliably predict the amount of free disk space that might be required during training, which lead to many interrupted trainings due to our servers running out of disk space. Also, the training process appeared to perform an excessive amount of disk input-output (I/O) operations, which lead to significant slowdowns exacerbated by the particular server architecture we use at our company.

These issues lead us to embark on an initiative to improve the Moses training toolkit to reduce the number of I/O operations and the peak disk usage. As a starting point we took a Moses release from mid-2010, as we considered it the most stable at the time.

The improvements we introduced were focused mostly on avoiding the generation of temporary files during the training process unless absolutely necessary. Where two tools could not directly talk to one another, we used UNIX-style named pipes to handle the data flow, which significantly reduced peak disk usage.

Finally, we noticed that a number of the training steps are independent of one another and could be run in parallel. We exploited this feature by modifying the training script (`train-model.perl`) to run the relevant training steps in parallel. The resulting memory-based data flow during the parallel execution of training steps is shown in Figure 1.

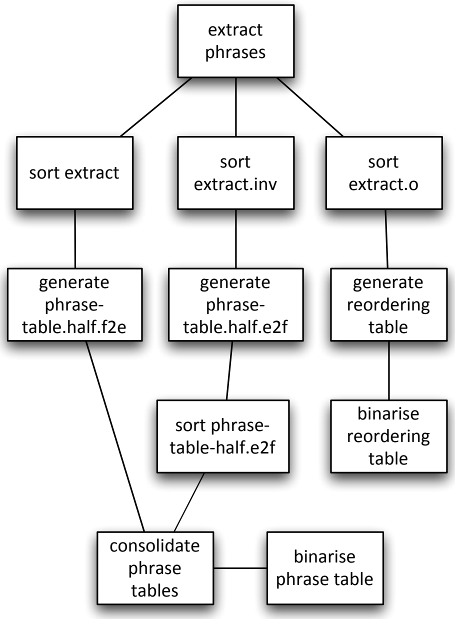

extract phrases
sort extract
sort extract.inv
sort extract.o
generate phrase-table.half.f2e
generate phrase-table.half.e2f
generate reordering table
sort phrase-table-half.e2f
binarise reordering table
consolidate phrase tables
binarise phrase table

Figure 1: Data Flow for the Parallel Steps of the Moses Training Workflow

A comparison of the peak disk usage and I/O operations during the training of an EN→JA engine with the original and improved workflows is shown in Table 1.

	Original Workflow	Improved Workflow
`extract` file size	7,5GB	uses pipe
`phrase-table.half` size	1,7GB	uses pipe
`phrase-table` size	2GB	uses pipe
`reordering-table` size	2,5GB	uses pipe
total disk I/O	196GB	23GB
peak disk usage	45GB	12GB
disk usage after training	9GB	6GB

Table 1: Disk Usage Statistics for EN→JA MT Training

The modifications to the Moses training toolkit listed above were provided to the MosesCore FP7 project for merging with the main Moses tree.

2.2 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 Perl-based and interact both internally and externally over TCP.

The first element 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 decoder for the requested language, together with any necessary pre- and post-processing tools. The MT servers read data one segment per line and output translations as soon as they are available, with the communication occurring over TCP. For each language that we use in production, we currently have 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 any 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 up to twenty 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 ease-to-use API for communication with the MT Info Service clients.

During the last twelve months, the MT Info Service received over 180 000 translation requests that were split into almost 700 000 jobs for load balancing. Among these requests were over one million documentation segments, as well as a large volume of UI strings.

2.3 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 localisation and SDL WorldServer for documentation localisation.

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, however, does present us with its own specific issues to handle — with its use of placeholders (PHs) to mask XML tags. The majority of our software documentation is authored using DITA-based XML and one goal of WorldServer is to hide the XML tags from the translators, as they do not represent actual content. The first issue here is that WorldServer only stores the PHs in the TMs and not the actual content they mask. For example, the segment

The <b>new</b> features of AutoCAD <ver/> are:

will be stored as

The {1}new{2} features of AutoCAD {3} are:

Please note, that any PH may be either an opening or closing formatting tag, or a standalone tag with or without semantic meaning in the structure of the sentence.

An major issue is that in the TMs the PHs are stored with IDs numbered by segment, i.e. in each segment the PHs start from 1; while during translation, the PHs are numbered continuously for the whole project, sometimes reaching IDs into the thousands. This means that any PH with an ID above about 40 will be treated as an unknown word, thus adding significant penalty during translation. We avoid this issue by temporarily renumbering PHs during translation making sure that — for any segment that the MT engines see — the PHs start with ID 1. The original IDs are then restored in the MT output. We found out that, with this process, our MT engines produce very little errors in the placement of PHs and we do not expect to achieve better performance by, say, first removing the PHs and then using word and/or phrase alignment information to reinsert them in the target.

Finally, as most PHs mask formatting XML tags, the whitespace surrounding the PHs is significant. It, however, gets lost during tokenisation and could lead to errors that are hard to identify and fix for the translators. For this, we added an extra processing layer during MT that preserves to the largest extent possible the whitespace surrounding the PHs in the source, regardless of the output of the MT engine and detokeniser.

So far we perused in detail the complex MT infrastructure at Autodesk. The question that arises is if there is any practical benefit of the use of MT for localisation and how do we measure this potential benefit. We present our answer in the next section.

3 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.

This is already the third productivity test that Autodesk performs. 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 paper was mainly to confirm our findings from the previous tests, as well as to help us pick among several MT options for some languages, as well as 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 versus automatic edit-distance-based indicators.

3.1 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.

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 users manual for PhysX (a plug-in for 3ds Max).

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 engaged four translators — one each from our usual localisation vendors — for two business days, i.e. sixteen hours. We let our vendors select the translators as per their usual process.

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 (like 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 a modified version of Pootle (`translate.sourceforge.net`) instead, as it is easier to manage and provides typical CAT functionality.

3.2 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. From the remaining data, we averaged the productivity (measured in words per eight-hour business day — WPD) for translating from scratch, taking a specific average for each translator and product combination. We had to use these separate baselines, as the variation between individual translators, as well as between different products for the same translator, is very big.

Comparing to the thus established corresponding baselines, we calculated the apparent productivity delta for each segment that the translators post-edited. The calculated average productivity increase per language is presented in Chart 2.

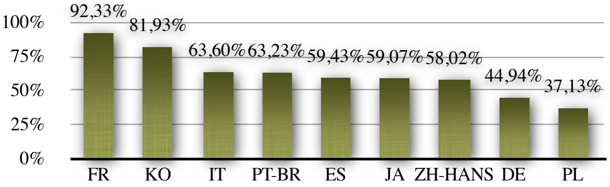


Chart 2: Average Productivity Increase per Language

A caveat is in order here. We need to point out that — due to the setup of our online workbench — we exclude certain translator tasks that are independent of the quality of MT from the productivity calculation. 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.

3.3 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 metrics we used were the following. METEOR (Banerjee and Lavie, 2005) treating punctuation as regular tokens, GTM (Turian et al., 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), CFS (Character-based Fuzzy Score, on tokenised data), WFS (Word-based Fuzzy Score). The Fuzzy Scores are calculated as the inverse of the Levenshtein edit distance (Levenshtein, 1965) weighted by the token or character

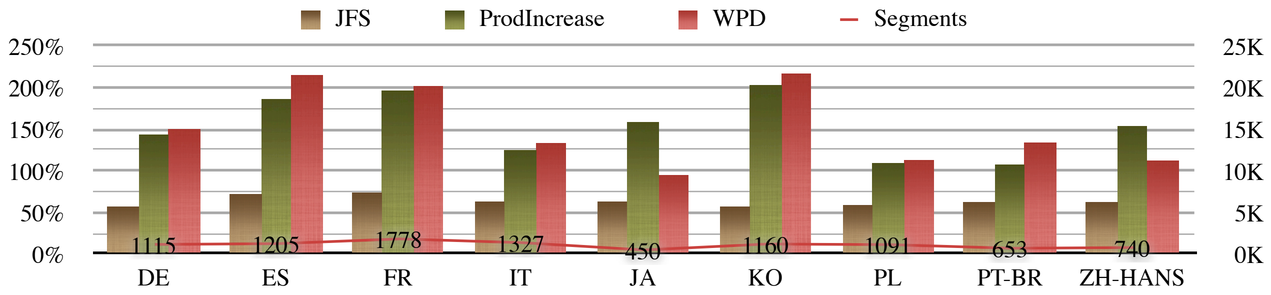


Chart 3: Edit Distance and Productivity Data for All Languages

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.

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 lumping the data for all languages together. We used Spearman's ρ (1907) and Kendall's τ (1938) as the correlation measures. The results are shown in Table 2.

	ProdIncrease	
	ρ	τ
JFS	0,609	0,439
SCFS	0,583	0,416
CFS	0,581	0,414
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
Length	-0,143	-0,097

Table 2: Automatic Metric Correlation with 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 translation. This suggests, for example, that a segment-length-based payment model for MT may not be a fair option. Also, we do not need to impose strong guidelines for segment length to the technical writers.

Considering the results, we decided to look for a possibility to create a joint metric that might exhibit 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). This metric 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. Chart 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 Charts 4–11, 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. 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. This also suggests that the productivity increase for KO shown in Chart 2 might not be realistic.

🟩 Maximum JFS 🟧 Average Productivity Increase 🟥 % of all Tokens 🟫 % of all Segments

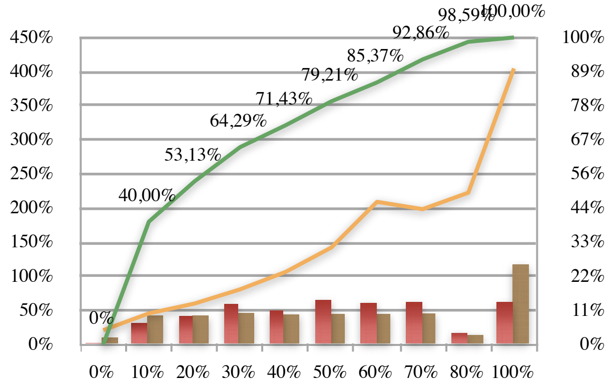


Chart 4: JFS to Productivity Correlation FR

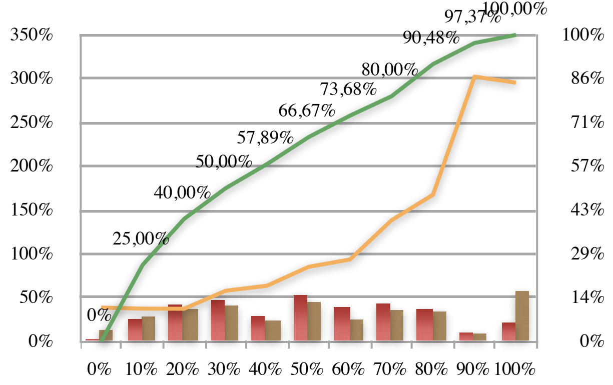


Chart 5: JFS to Productivity Correlation IT

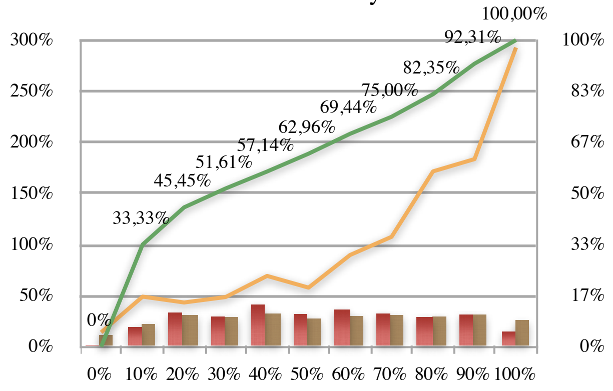


Chart 6: JFS to Productivity Correlation PT-BR

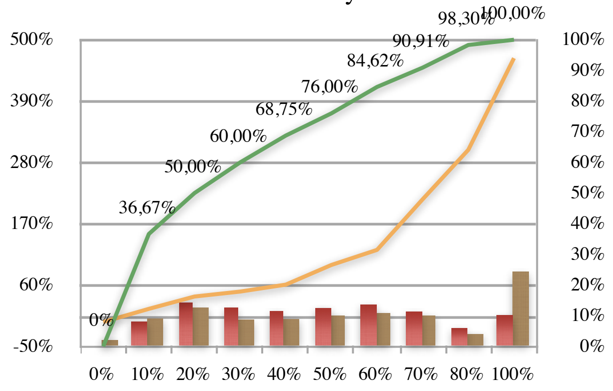


Chart 7: JFS to Productivity Correlation ES

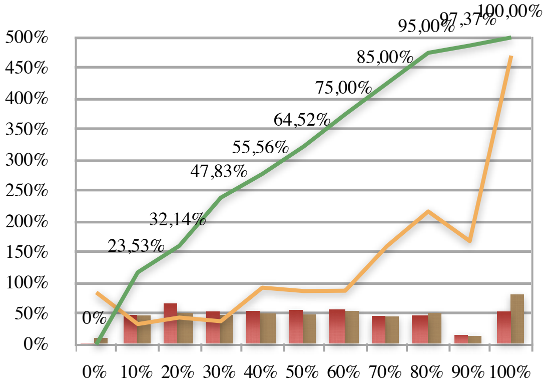


Chart 8: JFS to Productivity Correlation JA

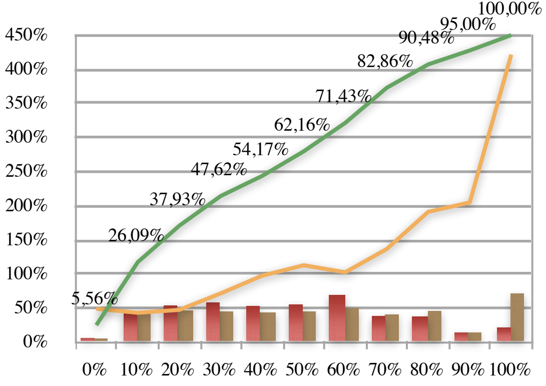


Chart 9: JFS to Productivity Correlation ZH-HANS

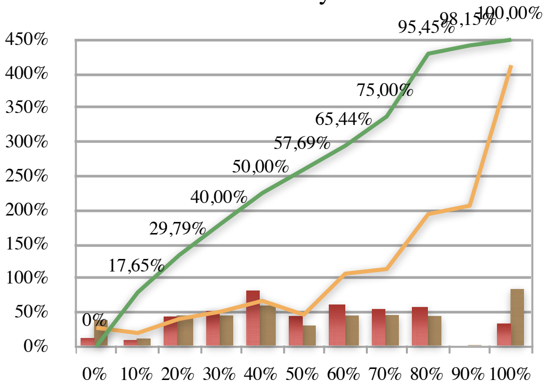


Chart 10: JFS to Productivity Correlation DE

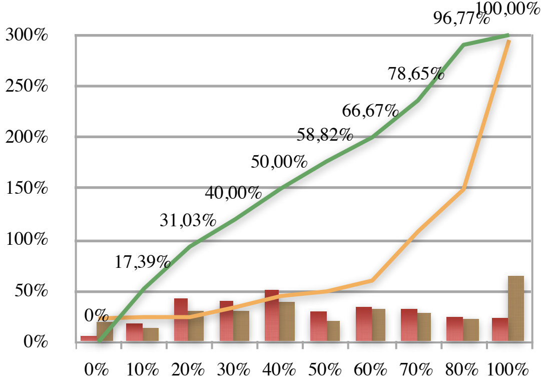


Chart 11: JFS to Productivity Correlation PL

A common observation for all languages is that both the worst and the perfect translations are predominantly short segments, which is as expected. First, it is much easier to achieve a perfect translation for a relatively short segment—especially given that JFS takes whitespace into account and our detokeniser is not perfect. Second, a complete rewrite of the MT suggestion usually results from an out-of-context translation of very short segments.

We also see that the JFS scores for the languages with the highest productivity increase (see Chart 2) are predominantly in the higher ranges, while for DE and PL there is a larger amount of segments with lower JFS.

In the next section, we try to apply the same evaluation methods to real-live post-editing data.

4 Evaluating Real-life Data

A new initiative at Autodesk, which will be extended significantly in the future, provided for the archival of all documentation segments that are post-edited in production. Currently, we store the EN source, the TM or MT target and the final target produced by the translators, but we do not have available any statistics on this data. In the future, we will store the original Fuzzy Match score from our TMs, as well as other metrics that we still need to decide on.

Of course, we do not have productivity data attached to the production segments, as our production environment does not provide for the aggregation of such data. Nonetheless, this is a wealth of post-editing data that we can analyse using the automatic metrics from Section 3.

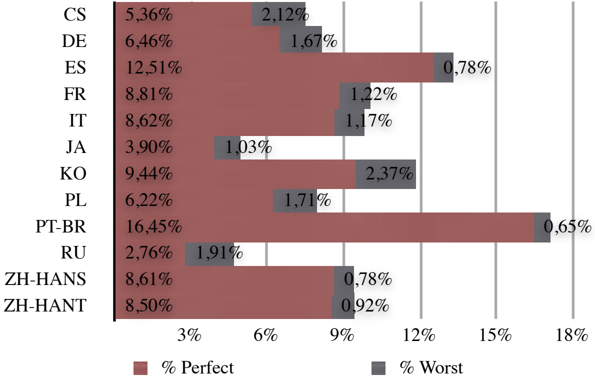


Chart 12: Proportion of Worst and Perfect MT

The first interesting piece of information is the proportion of worst and perfect MT translations, based on the performed post-editing. It is taken as the number of tokens in the worst / perfect translations versus all tokens for each language. Remember that only documentation segments that receive a fuzzy match score below 75% against our TMs are sent to MT. This statistic is presented in Chart 12.

The most important takeaway from this chart is that the proportion of worst translations is negligibly low. On the other hand, there are many perfect translations, despite the disadvantage of Machine Translating only those source segments that were not found in the TMs.

As a further analysis step, we can order the MT engines for the individual languages based on a specific metric per software product. The language order based on the derived JFS metric is presented on Chart 13 for the eight products with the largest translation volume.

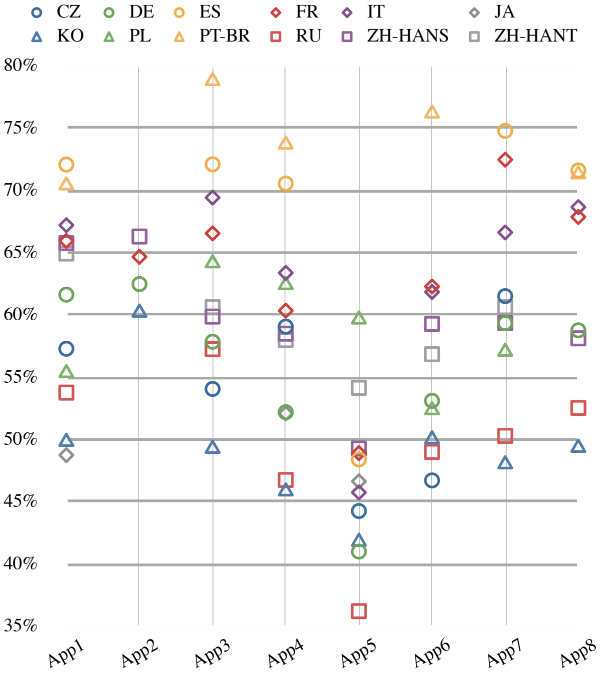


Chart 13: Language Order per Product according to JFS

Although this chart does not include data across all languages for all products, some trends are clearly visible. Namely, ES, IT and PT-BR often present the best JFS, while KO, JA and RU perform poorly on average. While we could expect lower quality MT for KO and JA, the data for RU need an extra explanation. In this case, the poor performance was due to a Unicode-related bug in the recaser for RU that was not detected until late in the production cycle. If we had analysed the data earlier on, we would have found the bug earlier on.

Another trend is for lower performance on average for App5. As it turned out, this was due to one single component within that product, for which the segmentation had failed and many segments contained new line characters. This could not be handled by the MT infrastructure and resulted in MT output that did not match the EN source.

We plan to integrate this type of analysis in a dedicated monitoring system, where we will automatically point our teams to potential issues with the localisation process. This will be accomplished by looking for suspicious patterns in the evolution of the JFS metric — a larger number of over- or under-edited segments may often be to either MT issues or translator under-performance.

For example, we are currently investigating the higher-than-average number of unedited PT-BR segments, given that there we have the smallest training corpus across all languages. We suspect that this could be due to translators leaving the raw MT output unedited without properly checking its correctness. This suspicion is also supported by the presence of a very large number of unedited Fuzzy matches for PT-BR.

5 Conclusion

In this paper, we described the MT infrastructure at Autodesk that is used to facilitate the localisation of software documentation and UI strings from English into twelve languages. We also investigated the data collected during our last post-editing productivity test and found a strong correlation between the edit distance after post-editing and the productivity increase compared to translating from scratch. Finally, we had a look at the post-edited data generated during production in the last twelve months comparing the MT engine performance for some of our products.

We plan to use the insights from the presented data analysis to continuously monitor the performance of our MT engines and for the (semi-) automatic detection of potential issues in the MT process.

References

Banerjee, Satanjeev and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgements. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*, pp. 65–72. Ann Arbor, MI.

Kendall, Maurice G. 1938. A New Measure of Rank Correlation [June 1938]. *Biometrika*, 30 (1/2): 81–93.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the Demo and Poster Sessions of the 45th Annual Meeting of the Association for Computational Linguistics (ACL '07)*, pp. 177–180. Prague, Czech Republic.

Levenshtein, Vladimir I. 1965. Двоичные коды с исправлением выпадений, вставок и замещений символов (Binary Codes Capable of Correcting Deletions, Insertions, and Reversals). *Доклады Академий Наук СССР*, 163 (4): 845–848.
[reprinted in: Soviet Physics Doklady, 10: 707–710.].

Neubig, Graham, Yosuke Nakata and Shinsuke Mori. 2011. Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT '11)*. Portland, OR.

Plitt, Mirko and François Masselot. 2010. A Productivity Test of Statistical Machine Translation Post-Editing in a Typical Localisation Context. *The Prague Bulletin of Mathematical Linguistics*, 93: 7–16.

Snover, Matthew, Bonnie J. Dorr, Richard Schwartz, Linnea Micciulla and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA '06)*, pp. 223–231. Cambridge, MA.

Spearman, Charles. 1907. Demonstration of Formulæ for True Measurment of Correlation [April 1907]. *The American Journal of Psychology*, 18 (2): 161–169.

Tillmann, Christoph, Stefan Vogel, Hermann Ney, Alex Zubiaga and Hassan Sawaf. 1997. Accelerated DP-Based Search for Statistical Translation. In *Proceedings of the Fifth European Conference on Speech Comunication and Technology (Eurospeech '97)*, pp. 2667–2670. Rhodos, Greece.

Turian, Joseph P., Luke Shen and I. Dan Melamed. 2003. Evaluation of Machine Translation and its Evaluation: Computer Science Department, New York University.