D3.3: Final Report on Interactive Editing

Philipp Koehn, Herve Saint-Amand, Vicent Alabau

Distribution: Public

CASMACAT
Cognitive Analysis and Statistical Methods
for Advanced Computer Aided Translation

ICT Project 287576 Deliverable D3.3

Project funded by the European Community
under the Seventh Framework Programme for
Research and Technological Development.

Project ref no.	ICT-287576
Project acronym	CASMACAT
Project full title	Cognitive Analysis and Statistical Methods for Advanced Computer Aided Translation
Instrument	STREP
Thematic Priority	ICT-2011.4.2 Language Technologies
Start date / duration	01 November 2011 / 36 Months

Distribution	Public
Contractual date of delivery	October 31, 2014
Actual date of delivery	January 7, 2015
Date of last update	January 7, 2015
Deliverable number	D3.3
Deliverable title	Final Report on Interactive Editing
Type	Report
Status & version	Final
Number of pages	33
Contributing WP(s)	WP7
WP / Task responsible	UEDIN, UPVLC, CBS, CS
Other contributors	Herve Saint-Amand, Turan Rustamli
Internal reviewer	Robin Hill
Author(s)	Philipp Koehn, Herve Saint-Amand, Vicent Alabau
EC project officer	Aleksandra Wesolowska
Keywords	

The partners in CASMACAT are:

University of Edinburgh (UEDIN)
Copenhagen Business School (CBS)
Universitat Politècnica de València (UPVLC)
Celer Soluciones (CS)

For copies of reports, updates on project activities and other CASMACAT related information, contact:

The CASMACAT Project Co-ordinator
Philipp Koehn, University of Edinburgh
10 Crichton Street, Edinburgh, EH8 9AB, United Kingdom
pkoehn@inf.ed.ac.uk
Phone +44 (131) 650-8287 - Fax +44 (131) 650-6626

Copies of reports and other material can also be accessed via the project's homepage: http://www.casmacat.eu/

Executive Summary

Contents

1 Overview

This deliverable reports on the work carried out in Workpackage 3, which aims at the development of new methods to assist the editing of translations. Such methods will assist human translators to improve machine translation output.

In the reporting period (month 25-36), work was carried out on the following task, as planned:

- Task 3.5: Display Multiple Translation Options
- Task 3.6: Authoring Assistance
- Task 3.7: Automatic Reviewing

We added one task: a study relating machine translation quality and post-editing effort.

2 Displaying Multiple Translation Options (Task 3.5)

This tasks developed two distinct types of assistance that suggest alternate translation choices to the translator.

- Translation Options in Context: The user may select any source phrase, and the assistance displays possible translations for this phrase in context, in form of a bilingual concordance.
- Translation Option Array: Automatic display of up to 5 different translation options for input words and phrases.

Work on the bilingual concordancer was concluded in the second year of the project. We introduced the concept of displaying multiple translation options to the translator in Deliverable 3.2 (Section 5.1) in the second year reporting period. This year, we integrated it into the CASMACAT workbench, and added some refinements to its user interface.

Figure 1 shows the basic display of the translation option array. Phrase translations of any length are ranked against each other. More likely phrase translations are ranked higher, and with a lighter background. The likelihood of a phrase, as layed out in last year's Deliverable 3.2, is based on the future cost estimate of the phrase translation (a combination of weighted phrase translation probabilities and language model costs), and the outside cost of the context.

2.1 Backend Integration

Translation options are provided to the graphical user interface via the CAT server, which receives them from the machine translation engine. The translation options are possible phrase translations that have been considered during the decoder search.

Note that the set of translation options may go beyond what is contained in the decoder's search graph, which is used by interactive translation prediction. The search graph is the result of heuristic search with several points of pruning (selecting only some of the translation options, removing dead end paths in the search, etc.).

Since the translation options do not change during the translation of a sentence, they are loaded upon start of a sentence translation. The translation option array is arranged by the graphical user interface.

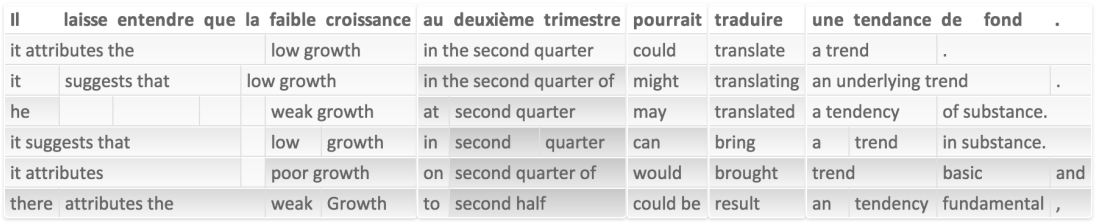


Figure 1: Translation option array

2.2 Interactions

One important purpose of the translation option array is simply its presents. By displaying alternate translation choices visibly to the user, they non-intrusively provide visual cues to the translator.

But the translator can also directly interact with the translation option array. By clicking on any of the target phrases, they are placed in the edit area. It is even possible to create a translation by clicking though the array. Translating becomes a form of puzzle solving.

2.3 Slider Display

For longer sentences, the translation options do not easily fit under the editing box. Our initial implementation, and similar efforts by other tools, arrange the translation options to sprawl over multiple rows. This has the disadvantage of placing information about the end of the sentence below the visible area of the screen.

In a refined setup (not yet used in the field trail, but in the community translation platform integration), we placed all options in one wide table below the textarea. Now, when we are dealing with a long sentence, where the table becomes too big, the translator can move the content with a slider controlled by the mouse.

2.4 Automatic Orientation

Having to manually slide the translation option display creates additional effort of the translator, which may be too burdensome and lead to disuse. The most relevant part of the display are the parts that are relevant for the words to be translated next, or just translated. So, the translation option display should be centered around these words.

Given word alignment information, we are able to do this. See Figure 2 how it plays out. Given that we predict that the next word to be translated is *erupted*, so we highlight that word in the translation option display and center the translation option array around it.

With this addition, the translation option array elegantly fulfills one of the core requirements of good user interface design: it does not require not unnecessary user interactions and displays the most relevant information given the current user needs. The user can always just glance below the editbox to learn about possible alternative translations.

2.5 Evaluation

We evaluated the bilingual concordancer in the third year's field trail. Please refer to Deliverable 6.3 for details.

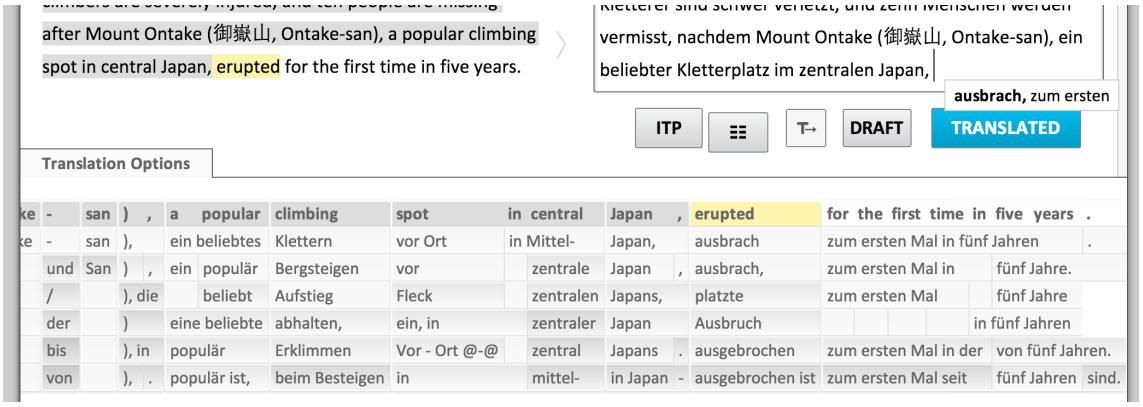


Figure 2: Automatic orientation of the translation option array: Given the source words already translated, the slider orients the next word to be translated into the center.

2.5.1 Qualitative Evaluation of Translation Option Array

We integrated the translation option array in the Community Translation Platforms integrations of the CASMACAT workbench. We did not formally evaluate the usage of this type of assistance, but translated commented on it in their feedback.

One translator commented

> *Die abschnittsweise Übersetzungen fand ich sehr übersichtlich und auch das System, wie die Sätze unten zerlegt wurden, fand ich sehr hilfreich und übersichtlich.*
>
> *Wollte ich jedoch ein übersetzes Wort mit einem anderen aus der Tabelle unten austauschen, wurde dieses leider nicht dort eingefügt wo mein Curser stand. Ich versuchte also das zu ändernde Wort zu markieren, vielleicht würde es so ausgetauscht werden, aber auch das gelang mir nicht. Das Wort kam immer ans Satzende und ich musste es dann mit Copy&Paste an die richtige Stelle setzen.*
>
> **Translation:**
> *I found the partial translations very clearly arranged, and also the system that breaks up sentences below, was very helpful and clear.*
>
> *However, when I wanted to replace a translated word with another from the table below, it was not placed at the cursor position. I tried to mark the word to be changed, in the hope that it would be exchanged, but that did not work. The word was always placed at the end of the sentence, so I had to move it with copy&paste to its correct position.*

This was not a usage that we anticipated, but it is very intuitive and should be added.

2.5.2 Quantitive Evaluation of Translation Option Array

A quantitative evaluation of the translation option array is not straightforward, since its utility does not only derive from direct interactions with clicks to target language phrases. A more common use may be that translator glance at it to get ideas about alternative word choices that are more fluent in context, but not part of the immediate active vocabulary of the translator.

Obviously, the translation option array should increase translator productivity, especially by reducing medium size pauses of 30-60 seconds, in which a translator consults an external dictionary. We could measure these.

Another measure would be the impact on the resulting translations. We would hope that a translator with access to this type of assistance would use a more diverse vocabulary of more finely tuned word choices. We could measure this diversity of the vocabulary with standard means, such as entropy of word distributions, or count of unique tokens.

3 Authoring Assistance (Task 3.6)

Our work on authoring assistance focused on the further development of our paraphraser tool, which allows the user to query the system for rewordings, in the same language, of any given phrase. This has several use cases: it can be used by a translator to find inspiration in selecting the best possible translation for a given input phrase, or it could also be used as a tool to quickly fix up an automated translation: since automated translations often contain some phrases that are not entirely out of place, but not quite the best selection for the given context, our tool should allow the translator to select, from the translation, a phrase which needs rewording, query the server for paraphrases, and, find from the list of suggestions a phrase that matches the input text better. This would allow the translator to speed up their work by reworking a translation without even having to type at all.

The work carried out for this task in this year involved testing and fine-tuning the core paraphraser algorithms, which had already been sketched out in the previous years, then building a Web service which provides a paraphrasing back-end for any Web-based application, and finally integrating that service as a component into the CASMACAT workbench.

3.1 Paraphraser algorithm

Our work in finding the best paraphrasing algorithm draws upon and extends the work that was carried out as part of CASMACAT in the first and second years.

The central data structure that is used to produce paraphrases is the phrase table. This data structure is of core importance to statistical machine translation, and hence is very well studied. The phrase table's ubiquity in SMT also means that there is plenty of well tested and highly performant software available to build, manipulate and query them. This is a major advantage and is one of the main reasons for selecting phrase tables as the core data structure for paraphrasing.

The phrase table is, in its essence, a bilingual dictionary. Its entries are phrases (of varying length), and translations are listed for each phrase along with a probability which indicates how frequent the translation is.

The basic idea behind the approach we took to paraphrasing is as follows: we build a translation system, configured to translate between language that we want to be paraphrasing for and some other language – the identity of the other language doesn't matter, although it helps to pick a language for which good resources and large amounts of training text are available. For our experiments, for instance, we built a system that provides paraphrases in English. Because we have a lot of resources available for translating between English and French, we used French as the second language. A system was therefore built to translate from English to French, and also another for translating back from French into English. We now had two phrase tables, one for each direction.

Then, to pharaphrase a given piece of English text, we first query the English-French phrase tables; this gives us various French phrases that are potential translations of the original. The trick is then to translate each of these back into English. Because each phrase has a high number of possible translations, by looking at all possible French translations of an English phrase and then all the possible English translations of each of these French phrases, we get a high number of potential paraphrasings. And since each translation has a probability associated with it, by multiplying the probabilities attached to each direction of translation we can assign each of the resulting paraphrases a probability, which allows us to rank them from best to worst.

From here the paraphrasing algorithm follows the same sort of algorithms that are typically used in statistical machine translation: we build a table that is filled with the best partial

rephrases, obtained in the process just outlined, and then search through that table to find the best combination of phrases that fully and exactly covers the input text. A language model is also used to ensure that the output is not only a likely rewording of the input but also fluent English. The algorithm is the same as is commonly used in SMT: the language model does not simply pick the best-sounding individual phrases, but pushes the phrase selection towards a combination of phrases which, when assembled end to end, form a fluent sentence.

This process is further streamlined by building a new type of composite phrase table, which we call a paraphrasing table, which is compiled from the two phrase tables used in the above steps. The idea is to precompute the multiplication step by essentially selecting every English phrase, then its French translations, and then the English translations of those. This creates an explosion in the number of translations, of course, so intelligent pruning is required to keep the number of paraphrases manageable; without pruning the paraphrasing table quickly grows to occupy terabytes of memory, mostly filled with low-probability rephrasings that would in all likelyhood not produce very good translations.

3.2 Paraphraser service

Once the algorithms were in place, we focused our attention on building a Web service to be used as the backend for our workbench. This is an HTTP-based API that receives simple queries for rephrasing, and runs the algorithm just described, and returns results in JSON format, which has the advantage of being easy to read both for computers (who will then have to display those paraphrases to the user) and for humans (and in particular the programmers who have to troubleshoot the system).

The reason for implementing the paraphraser as an independent piece of software is two-fold. First, it allows easier reuse of the rephraser as a software component. Other research groups might also want to make use of a rephraser, but they might not be building their system using the same technologies (PHP, Apache, Web Sockets) that CASMACAT uses. By having the rephraser on a completely separated platform, as long as the frontend can issue HTTP requests (which nearly any programming environment can), it can query the rephraser.

The second motivation for the separation is more practical: the rephraser can require rather large amounts of computer memory in order to store its paraphrase table in RAM. These requirements can be eased, but only at the cost of a proportional degradation in the quality of the output – lower-probability phrases can be removed from the paraphrase table, which frees up space, but the system is then working with poorer data. Such tuning of the system required a lot of testing and debugging, as we strove to find the right balance between resource usage and good performance.

The completion of the Web service API involved some technical challenges, mostly related to coping with multiple simultaneous requests. The software used to query the paraphrase table, which had been written prior to this project, and is used to query regular phrase tables as they are commonly used in SMT, had to be adapted to the particular requirements of running a Web server. Some parallelization techniques, in particular binary semaphores and messaging queues, were employed to work around the limitations of the original software.

With these hurdles passed we now have a working Web service API that can be easily redistributed and reused for other projects. The API, which is available in the public CASMACAT CAT server source code repository[1], should be easy to set up for anyone familiar with the usual menagerie of Moses tools for statistical machine translation.

[1] `https://github.com/hsamand/casmacat-cat-server/`

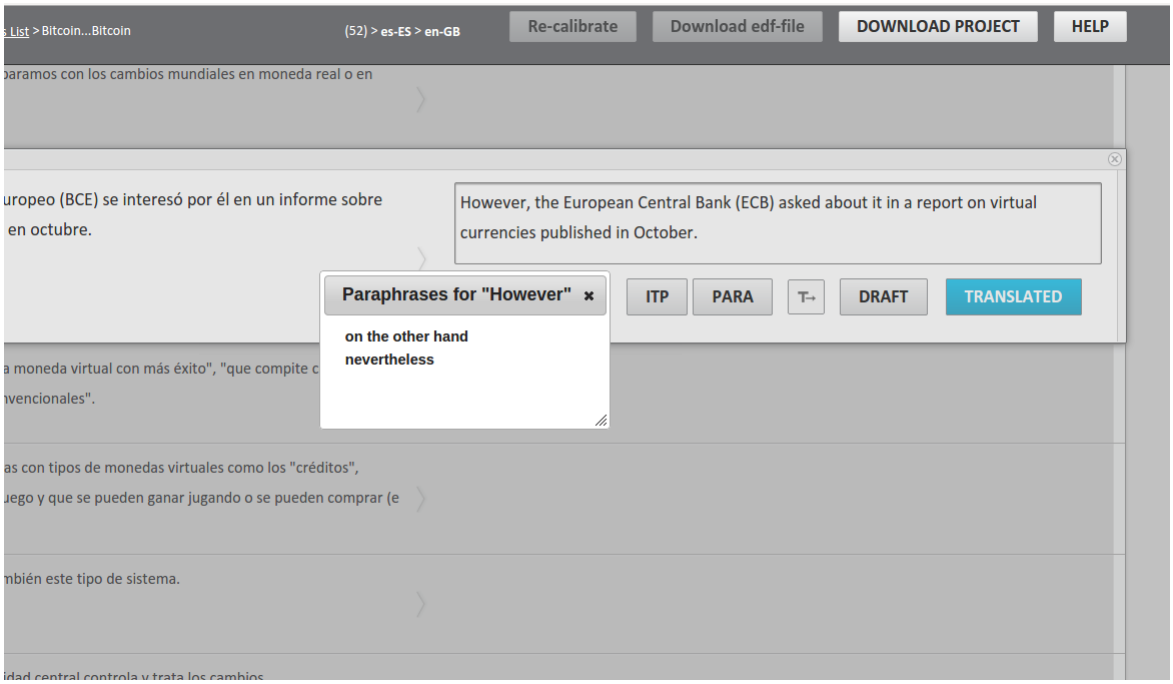


Figure 3: The rephraser in action: the user has selected the phrase “on your own”, and clicked “rephrase”. The system responds with a list of possible rephrasings (two, in this case: “alone” and “by yourself”).

3.3 Workbench integration

Once the API was working the final task was to integrate it into the graphical user interface of the CASMACAT workbench. This was a fairly straightforward task, as the presentation follows the same look and feel as has previously been used in CASMACAT, and so the main GUI components could be reused. The rephraser does connect to a different type of backend than the other components (it connects to the Web service API just described, over plain HTTP, rather than to the CAT server, which is reached using Web Sockets), but that didn’t add significant challenges.

We did consider making the rephraser part of the CAT server, for a more homogeneous network setup. This would simply mean that the CAT server, which the frontend already connects to, would accept requests for paraphrasing, which it would simply forward to the paraphrase API. The response would similarly simply be relayed to the frontend. Effectively the CAT server would serve as proxy server for the rephraser API. While this setup does allow a simpler network configuration, since the frontend only has a single endpoint to connect to, it did not appear to us that this would provide a significant improvement, and it can only reduce performance by adding an extra node for the data to hop by.

While we did not examine the utility of the paraphrasing assistance to translators in the CASMACAT workbench, we provided this service to Symantec within the ACCEPT project, who carried out extensive testing in the context of monolingual post-editing.

4 Automatic Reviewing (Task 3.7)

In this task we study how the reviewing process could be done more ergonomically and efficiently. First, introduce e-pen reviewing as a tool to help reviewers. Knowing that handwriting is an input method that is less productive than keyboard or speech, i.e., handwriting produces less characters per second, the use of the e-pen was only considered in a reviewing scenario where the user is likely to introduce less changes and can benefit from some conventional editing gestures

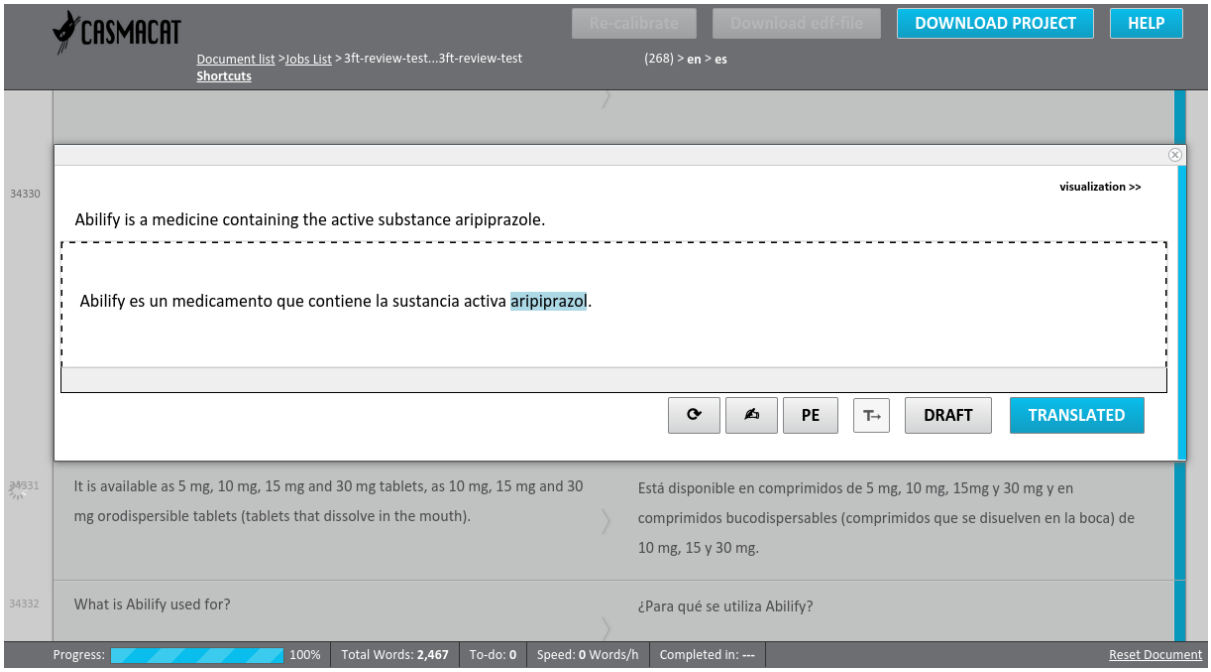


Figure 4: Screenshot of the CASMACAT workbench adapted to e-pen. Note the vertical layout and drawing are surrounded by dashed lines.

and improved ergonomics. Thus, one of the aims of the third field trial was to assess how professional reviewers use the e-pen functionalities implemented in the CASMACAT workbench. Second, we study how to automatically detect and correct the errors by learning how the reviewers performed the corrections during the field trials.

4.1 Reviewing with the e-pen

Intuitively, e-pen can be more ergonomic than mouse and keyboard when producing few text changes since a reviewer proceeds as if she was operating with a paper and a pencil. To enable e-pen reviewing in the CASMACAT workbench we used the user interface developed in WP5 (Task 5.3) and leverage the research carried out in WP2 (Task 2.2).

On the one hand, the frontend was a modified version of the CASMACAT workbench in which the horizontal layout was replaced with the a vertical layout, with the purpose of providing a bigger drawing area with bigger fonts (Figure 4). The frontend was also responsible of decoding gestures using MINGESTURES (Leiva et al., 2014).

On the other hand, the backend was an On-Line Handwritten Text Recognition (HTR) server with iAtros technology (Luján-Mares et al., 2008), which was responsible of decoding the handwritten strokes into text. Stroke preprocessing and feature extraction was achieved by using a standard procedure (Toselli et al., 2007). For morphological modelling, continuous density Hidden Markov Models (HMMs) were trained with the the IBM-UB-1 database (Shivram et al., 2013). The EMEA corpus was used to train 2-gram language models, with a 5,000 vocabulary. For more details see on how the system was trained see subsection 4.5. In a second round, the iAtros server was replaced a commercial HTR system.

4.2 E-pen evaluation in the third field trial

The third field trial involved seven post-editors and three reviewers (see D6.3). All post-editors and reviewers were freelancers recruited by Celer Soluciones SL. After the seven translators

in the CFT14 post-edited the two texts, three different reviewers proof-read their final target texts. Reviewers' profiles in CFT14 are presented in Table 1. Each reviewer was assigned a series of translator whose translations were reviewed under two different conditions:

- *Condition 1*: Traditional revision (R), i.e. using the keyboard as the only input method.
- *Condition 2*: Revision using an e-pen(RE), i.e. using an e-pen as an input method to enter corrections in the text.

Table 1: Reviewers' profile in CFT14 study.

Participants	**R01**	**R02**	**R03**
Gender	M	F	M
Years of translator training	4	4	5
Reviewing experience	Yes	Yes	Yes
Years of professional experience	8	3	31
Have you ever used an e-pen?	No	No	No
Translators reviewed	P01, P02, P03	P04, P05	P06, P07

We performed two rounds of e-pen reviewing. In order to improve recognition rates, in the first round the three reviewers trained the system handwriting 100 different words prior to the experiment, which were used for writer adaptation (Martín-Albo et al., 2014). In this initial phase, reviewers were also instructed on how to use the e-pen window in CASMACAT as well as on which gestures they could use for reviewing purposes, i.e. deletion, insertion and undo gestures. When performing under the e-pen condition, reviewers were asked to try twice in case handwriting recognition was not successful. If recognition failed in a second attempt, they could use then the keyboard to enter their corrections before continuing with the revision process using again the e-pen.

Unfortunately, after the first round of experiments, reviewers reported unusually high handwritten recognition error rates. As this fact could have affect users' perception regarding the e-pen system, we decided to repeat the experiment in a second round using a commercial handwritten text recogniser, which we found to be more accurate.

4.3 Reviewers' feedback

We collected reviewers' feedback through talk-aloud interviews. We started each interview with an informal conversation where we asked the reviewer about her subjective perception of the recognition accuracy of the e-pen system. In the first round of experiments, users estimated that e-pen recognition accuracy was about 50%. This ratio increased to an 80% recognition accuracy in the second round of experiments. After this informal start, we continued collecting reviewers' feedback through these three basic questions:

1. How would you evaluate revision using an e-pen: 1 (*very dissatisfying*) - 5 (*very satisfying*)?
2. According to your own personal opinion, what are the advantages and disadvantages of using an e-pen for revision (as compared to using a keyboard)?
3. How would you suggest to make the e-pen interaction in CASMACAT more successful (more productive)?

Two out of the three reviewers in the study evaluated the used of an e-pen for reviewing purposes as 2 (dissatisfying), whereas the third one rated it as 3 (neutral). It would be interesting to see if their views become more positive after some more hours of interaction with the e-pen.

When asked to enumerate possible advantages and disadvantages of reviewing using an e-pen, all participants mentioned recognition errors as the main disadvantage, together with the time spent to handwrite their corrections. "Despite 100% successful recognition using an e-pen, I think I will never be as fast as typing", mentioned R02. On the positive side, reviewers found some the the e-pen gestures (e.g., delete words) particularly useful. Interestingly enough, none of the three reviewers mentioned the ergonomic benefits derived from being able to switch from the keyboard to the e-pen at their convenience.

As a way to make e-pen interaction more successful, R01 suggested to make the e-pen window bigger in order to have more freedom of movement with the e-pen. The same participant also suggested to increase the separation between words in the e-pen reviewing mode in order to make the gesture for insertion more accurate and less error-prone. R02 and R03 mentioned that user experience can be improved if the e-pen could be directly used on the screen instead of having to use an external pad to handwrite while looking at the computer screen.

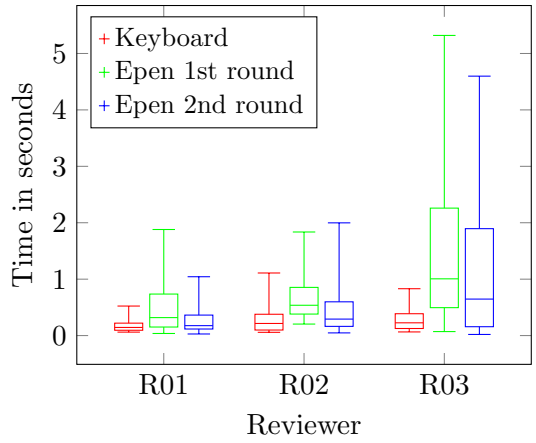
In the second round, the users noticed that the handwriting recogniser was more accurate. However, they did not change their opinion with respect the e-pen as a tool for reviewing.

4.4 Reviewer's activity data

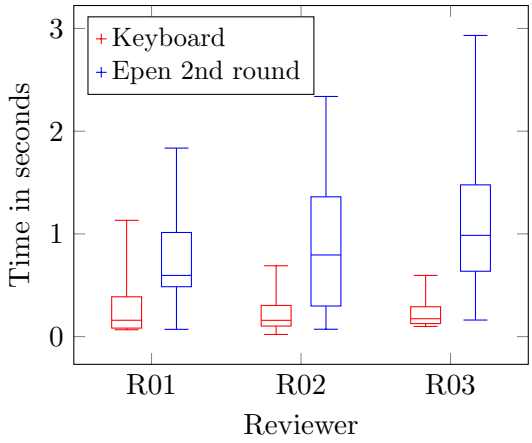
Regarding the analysis of the reviewer activity data, Figure 5a shows the boxplot for the number of seconds (normalised per source character) that took to review all the documents for each reviewer. Only the segments where the reviewers modified the translators' text have been considered. Time is counted from the moment the reviewer opened a new segment up to the moment the segment was closed.

First of all, we can observe that reviewers spent more time in the first e-pen round than in the second. As we have explained, this is due to the fact that the recogniser committed more errors and, thus, reviewers had to spend more time retrying to introduce the correct text. Second, we can observe that, for reviewers R01 and R02, e-pen reviewing was just a bit slower than in condition 1. Although we had expected that for small number of correction e-pen could be faster, it was also expected that handwriting would be slower than typing. In addition, in handwriting one must wait for the server to respond. In average, the server took $355ms$ to return the answer, whereas in 95% of the cases is was lower than $800ms$, which could explain part of the difference between keyboard and e-pen. Also, we should note that e-pen only allows to work at word level at this moment. However, keyboard interactions can be at subword level. In fact, we observed that `aripiprazole` was frequently replaced by `aripiprazol`. When using the keyboard, the correction consisted in just one mouse movement and click followed by a character deletion with the keyboard. Conversely, the whole word had to be written with the e-pen, and in case of recognition failure, it should be rewritten. Taking all that into account, the differences between keyboard and the second round of e-pen are actually small for R01 and R02. We think that an improved UI and text recogniser with longer user training would result in a very similar performance for both systems. In contrast to the other reviewers, R03 performed very bad with the e-pen. R03 was not only the reviewer with most experience (by large) but also the one who complained most about the e-pen system limitations. Thus, we deduce that for some reason R03 did not get used to that kind of interaction.

One of our assumptions was that working with the e-pen should allow for starting with the corrections earlier. Thus, for each segment we computed the time from the moment the segment was opened to the first interaction. We report on then medians instead of the means since they are more robust to measure the central tendency. The differences in the medians



(a) Number of seconds that reviewers spent on each segment normalised by source character. The time accounts for the total time spent in the segment from the moment it was opened until it was closed. Segments that were not modified are not considered in this plot.



(b) Number of seconds that reviewers spent on each interaction normalised by the number of characters edited in such interaction. Interactions refer to complete sequences of keyboard, mouse, and e-pen actions that take to accomplish a correction, including e-pen retries and fixing typos introduced with the keyboard.

indicate that R01 took 302ms more with e-pen than with keyboard, whereas R02 and R03 used 520ms and 3s more, respectively. As expected, R03 was the one who performed worse since he did not get used to the digitaliser pen. Furthermore, the digitizer pen used in the evaluation is arguably more difficult to use than the mouse for those who face it for the first time. Probably, a pen on a tactile screen would have been a much better choice since users can directly place the pen where they have their sight fixated. This fact was already acknowledged by reviews in the reviewers' feedback. Previously to the evaluation we had argued whether this could be a problem or not, and we had arguments in favour and against both, tablet and digitiser pen. Indeed, we had performed an early test with a Lenovo tablet with pen technology. However, that suggested that the CASMACAT interface was too resource hungry for that kind of devices. Thus, a digitizer tablet with a desktop computer was our only choice. In the light of these results and the reviewers' feedback it seems clear that a specific UI should be developed for tablet devices.

In a further step, we counted the median of the time it took to make a correction per each character produced, trying to isolate text introduction from the cognitive process. We also considered the time consumed in the retries. The results are shown in Figure 5b. As it was expected, text production with e-pen is a bit slower than with keyboard, roughly between 500ms and 1s. When we looked at the number of retries, we observed that in condition 1 reviewers retyped parts of text in 7% (R01), 11% (R02), and 2% (R03) of the interactions, usually involving the correction of just a few characters. However, in condition 2 reviewers changed their initial correction in 28% (R01), 26% (R02), and 16% (R03) of the interactions. In these cases, reviewers either rewrote the whole word or issued a delete or undo gesture, or typed the correction with the keyboard.

4.5 Closing the gap between iAtros and the commercial system

In the previous subsection, we have stated that our initial on-line HTR system did not performed as well as expected. Therefore, we decided to create a corpus from the e-pen evaluation in the second round to run lab experiments that could identify the problems with the iAtros setup. Thus, we analyse and compare here the performance of iAtros and the commercial HTR system in order to close the gap observed during the evaluation. In consequence, the e-pen data captured in the second round of e-pen reviewing was manually annotated and thereby built into

an experimental corpus. Using this corpus we report results of using different technologies in order to improve the accuracy of the e-pen interface.

It should be clear that, in contrast with the traditional keyboard and mouse, the e-pen interface is error-prone. Therefore, to make this interface usable for translation-editing purposes its accuracy has to be improved as much as possible. This motivates the introduction of novel technologies to try to improve e-pen accuracy as much as possible. Among these technologies, a recently proposed writer adaptation technique, which needs very little adaptation data, proves to archive a significant reduction in e-pen recognition errors. We summarise below the main parts of the HTR system used in the experiments.

4.5.1 On-line HTR system

In the following paragraphs, we detail the baseline on-line HTR system that was used in the first evaluation and the modifications that were necessary to improve the recognition accuracy.

Preprocessing and Feature extraction. The preprocessing of each e-pen sample involves four steps: size normalisation, repeated points elimination, noise reduction and writing speed normalisation (Toselli et al., 2007). Size normalisation is performed by re-scaling the point sequence to a fixed height, preserving the original aspect-ratio. Noise in handwriting is due to erratic hand motion and inaccuracy of the digitalisation process. In order to reduce this effect, we employ a smoothing technique consisting on replacing every point in the trajectory by the median value of its neighbours. Finally, the writing speed normalisation is performed to achieve an independence from the user writing style. This is implicitly carried out by first derivatives normalisation in the feature extraction phase.

Once the original coordinate sequence of each sample has been preprocessed, it is transformed into a new 6-dimensional real-valued time-domain feature vectors; namely: point location (y coordinate), first and second derivatives and curvature.

Morphological modelling. The morphological module is based on continuous density Hidden Markov Models (HMMs). A HMM is a stochastic finite-state device used to estimate the probability of a sequence of feature vectors, which characterise the time evolution of a given handwritten character. The IBM-UB-1 database (Shivram et al., 2013) has been used to train the HMMs. IBM-UB-1 contains free form cursive handwritten pages in English. It contains 6,654 pages of online data collected from 43 writers (4,138 summary pages and 2,516 query pages). Here we used the query partition which is formed by more than 64,000 isolated words.

Writer Adaptation. It is known that for a given handwriting recognition task, a writer-specific system will outperform a writer-independent system. But this statement is true as long as there is enough training data to obtain a good estimated writer-specific model.

If the amount of writer-specific training material is limited, however, such a performance improvement is not guaranteed. Under these conditions, one way to improve the system performance is to make use of the some multi-writer existing knowledge, so that only a minimum amount of writer-specific training data is sufficient to model the new writer. Such a training procedure is often referred to as writer adaptation.

Here a method for automatic generation of synthetic handwritten words based on the Kinematic Theory and its Sigma-lognormal model has been used (Martín-Albo et al., 2014). The objective is to produce different instances of the original word by simulating the intra-variability found in real human handwriting. To generate a new synthetic sample, first a real word is modelled using the Sigma-lognormal model. Then the Sigma-lognormal parameters are randomly

perturbed within a range, introducing human-like variations in the sample. Finally, the velocity function is recalculated taking into account the new parameters. The synthetic words are then used as additional training data for a Hidden Markov Model based on-line handwritten recogniser.

Language Modelling. The recogniser used in these experiments is word-based, which motivates the use of a closed vocabulary lexicon as will be discussed below. Word-based recognisers rely on a precompiled, maybe very large vocabulary of expected words. This kind of recognisers are know to provide superior accuracy; however they have the obvious drawback of being unable to recognize words which are not included in the vocabulary.

In this case, the EMEA corpus has been used for language model training. It consists of about 60 million words. Two different language models have been trained from this corpus:

LM1 A closed 2-gram language model: The underlying vocabulary consists of 5,080 words, which included the 5,000 most frequent words in training plus 80 unseen test words. The purpose of this language model is to make sure that all the words which are going to appear in the test set (i.e., all the words which the writers actually wrote) have a chance of being recognised. Most of the 80 words added in this way are rare words of specific medicaments. The 2-gram language model was trained taking into account the frequency of occurrence of the words which appear in training and it is uniform for the remaining words.

LM2 An open vocabulary 2-gram language model: which included the 2,000 most frequent words in training and the 5,000 most probably words from the IBM1 model given the source sentences, altogether 6,072 words. The rate of running out of vocabulary words is 15.4%. Therefore, this is the minimum error which could ever be achieved using a word-based recogniser.

The commercial HTR system. The commercial HTR system was used as a black box, and the technology behind it was unknown to us. Still, we could identify that the system was able to leverage a given prefix and it allowed char-based recognition. Hence, this system was able to recognise words that are not in a dictionary.

4.5.2 Experimental Set-up

The target was to recognize 266 samples obtained from the second round of the third field trial performed by 3 different reviewers. These samples contain handwritten text introduced in order to correct translation errors. Three different recognisers were used, including the commercial one used during the field trial proper.

4.5.3 Results

Results are shown in Table 2. When marked with "*", the results correspond to ignoring the case of words. In this case, results are better and the improvement with the writer-adaptive system is most significant. This is explained because the adaptation samples did not even include all the possible capital letter and, therefore, the adaptation to capital letter writing could not be as good as with the other characters.

On the other hand, since the recogniser is word based, in the open vocabulary experiments there is a minimum possible error of 15.4% because of the 80 test words which can never be recognised. Therefore, results should be considered better than they appear. In fact it is expected that many of the out-of-vocabulary errors can be recovered by using a word-based recogniser which can also work at the character level as a back off method.

Table 2: E-pen handwritten text recognition results. (*) Case insensitive.

System	Error
Commercial	14%
Commercial (*)	12%
LM1	
Initial result	22%
Writer adaptation	15%
Writer adaptation (*)	11%
LM2	
Writer adaptation	33%
Writer adaptation (*)	26%

4.6 Analysis of the Reviewers' Edits

In this subsection, we analyse the results from the reviewing process in the field trial to gain insight in how to perform the reviewing automatically. To prepare work on automatic reviewing, we first classify the types of edits done by the reviewers. Here, we did not draw a distinction between revisions done with the e-pen or with the keyboard.

We automatically detected edits between the draft translation and the revision (using the optimal string edit path over words), and classified each edit (changes affecting neighbouring words). In the revision data from the field trial, we obtain the following types of edits:

- 171 insertions — vast majority function words
- 152 deletions — about half substantial content
- 621 replacements — of which:
  - 75 changes to punctuation only
  - 28 change to lowercase / uppercase
  - 29 cases that are mostly deletions
  - 8 cases that are mostly insertions
  - 289 morphological changes (Levenshtein distance of less than 50%)
  - 190 other changes, about equal amounts function words and content words

How many of such errors could be hope to automatically detect?

Some of the edits may be due to spelling and grammatical errors. Since there are established methods for such type of errors, we do not want to focus on them here.

Insertions and deletions of function words will also be hard to detect. Instead, we want to focus on insertion and deletion of content. We assume that the reviewer inserts content, because there was content in the source sentence that is missing in the translation. Conversely, we assume that the reviewer deletes content, because it is not warranted by the source sentence.

Replacement of content words are another challenge (again, we do not attempt to improve on punctuation changes, morphological changes, or changes to function words). If a reviewer replaces a content word with a word of similar meaning, but with higher semantic and pragmatic plausibility, then it is doubtful that automatic methods given the current state of semantic processing would be able to predict that. The only type of content change that we hope to detect at this point, are changes due to inconsistency in choice of terminology. If a source word occurs multiple times in a single document, then we would expect it to be translated the same way into the target language.

In summary, automatic reviewing should detect:

- insertion of content words
- deletion of content words
- replacements to enforce consistency of terminology

4.7 Automatic Reviewing

Our methods are based on automatic word alignment of source sentences and their draft translations. We may use the incremental alignment methods developed by the CASMACAT project, or simply add the source/draft sentence pairs to a baseline parallel corpus and process them with standard word alignment methods. In our experiments, we used the latter approach, employing GIZA++.

Given word alignments between source and draft translation, we can detect:

- insertion of content words: any sequence of words in the draft translation that has no alignment point to the source
- deletion of content words: any sequence of words in the source sentence that has no alignment point to the target
- replacements to enforce consistency of terminology: any source word that occurs multiple times and is aligned to different words in the draft translation

Since we want only address changes to content words, we ignore words that have less than 4 characters, which is a reasonable assumption for languages such as Spanish, English, and German. In a future refinement, we could use part-of-speech tags as filter.

4.8 Evaluation

The evaluation of our automatic reviewing methods is severely hampered by lack of test data. To our knowledge, there is no available large corpus of manual revisions to human-produced translations.

Hence, we had to rely on the data generated by the CASMACAT project. The first data set is the described above: The revisions made in the third field trial. We also evaluate the methods on a second dataset, the volunteer translations from the community translation platforms. Especially the second data set is very small, so the results are just indicative of the potential success of these methods.

4.8.1 Evaluation of Field Trial Data

We did not find enough examples of inconsistent terminology in the field trial data, so we focused on the evaluation the detection of insertions and deletions.

Deletion: Target language words that have no alignment point in the source are expected to be deleted by the reviewer. We can compare if the prediction of a deletion action matches an actual deletion action in two ways: A strict way, where we predict that the word, say, *terapéuticas* will be deleted, and this word is indeed deleted in a reviewing action. Or in a more generous way, where we predict that there will be *some kind of* deletion in the sentence, and in fact the reviewer deletes a word.

	Strict Scoring		Generous Scoring	
Edit type	Precision	Recall	Precision	Recall
Deletion	7%	27%	11%	48%
Insertion	-	-	5%	35%
Any edit	-	-	20%	60%

Table 3: Automatic evaluation of the automatic reviewing method on data from the third year field trial. We measure how accurately we predict reviewing actions.

Insertion: Our method does not prescribe which words should be inserted, since it only detects source words that may not have been translated. So, we are only using the generous scoring method.

Any edit: Since some replacements may be rewordings that add or drop content, our evaluation may be too harsh, since we penalize prediction of deletion or insertion action, while the reviewer makes a replacement. So, we also score how often we predict any edit action. Note that this includes semantic changes, for which we currently have no hope to make correct predictions.

We only consider actions affecting content words. We also do not count as edit actions: morphological changes, or changes to lowercase/uppercase or punctuation.

Results are summarized in Table 3. Note that 14% of sentences were edited by the translator (7% have deletions, 4% have insertions). Our method is generally too eager, so recall is much higher than precision (we predict edit actions for 42% of sentences). Our performance is above trivial baseline methods (precision of 20% for any action, 11% for deletions, and 5% for insertions). More than half the time, when we predict a deletion successfully, we identify the correct word.

However, it is not clear, how meaningful these metric scores are to judge the viability of the method practical use. For this, we need to carry out manual evaluation of the method, which we do in the next section.

4.8.2 Evaluation on Community Translation Platform Data

We now subjectively evaluate our automatic reviewing methods on data obtained from the community translation platform installation of the project. We have the most data for the English–German language pair, so we carried out the evaluation on just this one language pair. We do not have manual revisions, so instead we judge the alleged mistakes found by the method manually. Note, that this way we only evaluate precision, not recall. We do not know, if we miss mistakes that the method should have detected.

In this data, the method correctly spotted two cases of deletions, one case of a possible[2] insertion, and one case of a possible deletion (error in bold):

`deletion:` *labour for* **Scottish Independence***; photo by Màrtainn MacDhòmhnaill, used under a CC by-NC 2.0 license.*
Labour für; Photo by Màrtainn MacDhòmhnaill, used under a CC by-NC 2.0 license.

`deletion:` **after** *congratulating Mahinour, I believe that her release from prison is a political ploy by an oppressive regime.*
ich gratuliere Mahinour, aber ich glaube, dass ihre Freilassung aus dem Gefängnis ein politischer Trick eines unterdrückerischem Regimes ist.

[2] annotator uncertainty

`possible deletion:` **would** *flow out to all the other parts of the body ,*
zu all den anderen Teilen des Körpers fließt ,

`possible insertion:` *Someone sure wanted people to know that he was thankful for Togolese President Faure Gnassingbé generosity.*
Ganz klar wollte jemand es **allgemein** *bekannt* **machen**, *dass er für die Grozügigkeit des togolesischen Präsidenten Faure Gnassingbé dankbar war.*

However, there were also 31 cases of false alarms, which can be broken down into the following cases:

Count	**Type**
16 cases	unaligned verb
6 cases	one-to-many alignment
2 cases	non-literal
6 cases	misalignment, often due to unknown word
1 case	valid verb ellipsis, repeated in sub clause

A notorious problem of German–English word alignment with basic methods such as the IBM Models is the high rate of failure to align the German verb. The verb is often highly reordered with respect to English, which causes problems with methods that have a bias towards monotonicity.

The other types of error are also intuitive given the imperfection of word alignment: whenever one source word should be aligned to two target words, then one of the target words may remain unaligned (or vice versa). The problem is even more acute in non-literal translations. GIZA++ word alignment has also a well-known problem with rare words (especially words that are unknown in the baseline corpus). Such words are often aligned to several words, leading to misalignments.

Overall, on this data set, we have a precision of 4 out of 35, which is just over 10%. This is likely too low to be useful. A translator would have to go through many false alarms, which will lead of frustration and lack of use.

4.9 Discussion

In this task, we have tackled the problem of the reviewer under two perspectives. First, we have studied the use of the e-pen to correct translator's mistakes. Second, we have analysed the reviewer's corrections in order to find patterns that could be performed automatically.

On the one hand, the evaluation of the e-pen as a reviewer tool has provided us with very valuable information. Firstly, we have discovered that e-pen interaction for reviewing needs a specific user interface that should be deployed in tablet devices and should be different (and lighter) from the post-editing user interface. Secondly, reviewers have shown their preference for a full-screen writing experience, as sometimes they found themselves writing on the border of the e-pen area. On the other hand, font sizes should be bigger and word separation wider so as to allow the user to be more precise when issuing gestures. Not only that, but the GUI should be redesigned to fit reviewers' needs. Furthermore, the handwriting recognition system must be improved since accuracy is one of the main reviewer concerns and the main factor why productivity suffers when compared to the keyboard. On the positive side, gestures were perceived as very useful, suggesting that a mixture of gestures and keyboard could be a good choice. However, to accomplish all these challenges we would have needed much more effort that

the one allocated in the scope of the CASMACAT project. Nevertheless, we are very optimistic that these issues can be overcome to create an efficient e-pen reviewing tool.

On the other hand, we have studied how to detect insertion, deletions and replacements of content words. Due to the lack of data, the experiments were carried out with a very small dataset. The results show that we can achieve a precision just over 10% according to subjective evaluation, which may be insufficient to be useful. However, this has been our first attempt to tackle this problem, from which we have obtained useful insight and ideas for future developments.

5 Machine Translation Quality and Post-editing Productivity

Intuitively, better machine translation quality leads to better post-editor productivity. Consider the extremes: If the machine translation output is useless, it can only lead to slower translation speed than translating from scratch, but near-perfect machine translation only requires quick polishing.

Given todays level of machine translation quality, how do differences between MT systems impact translator productivity? It is an open research question, how the quality increases measured by automatic metrics and subjective evaluation criteria relate to actual increases in the productivity of post-editors.

We carried out a pilot study to investigate the influence of the underlying SMT system on post-editing effort and efficiency.

We used four English–German machine translation systems that participated in the WMT 2013 machine translation evaluation campaign: Edinburgh's phrase-based and syntax-based systems, a popular online translation system from a large American internet company (anonymized as ONLINE-B), and one of the lesser performing system in the competition.

Nine different news stories, originally written in English, with a total of 500 sentences were post-edited by four non-professional translators, with machine translation output chosen randomly from any of the machine translation systems.

The results are summarized in the table below:

System	**MT Quality**		**Speed**	
	BLEU	subjective	**sec./wrd.**	**wrds./hr.**
UEDIN-SYNTAX	19.4	0.614	5.38	669
UEDIN-PHRASE	20.1	0.571	5.45	661
ONLINE-B	20.7	0.637	5.46	659
UU	16.1	0.361	6.35	567

While there is clear evidence that a 4 BLEU point difference leads to slower average post-editing speed (about 1 second per sentence, or about 20%), the differences of the other systems are too close in both machine translation quality and post-editing speed to make any grand sweeping statements.

It is worth noting that the syntax-based system which is known to not be properly appreciated by the BLEU score (note its much better subjective human judgement score), led to the fastest post-editing speed.

For more details of this study, please refer to Appendix A.

References

Koehn, P. and Germann, U. (2014). The impact of machine translation quality on human post-editing. In *Proceedings of the EACL 2014 Workshop on Humans and Computer-assisted Translation*, pages 38–46, Gothenburg, Sweden. Association for Computational Linguistics.

Leiva, L. A., Alabau, V., Romero, V., Toselli, A. H., and Vidal, E. (2014). Context-aware gestures for mixed-initiative text editing uis. *Interacting with Computers*.

Luján-Mares, M., Tamarit, V., Alabau, V., Martínez-Hinarejos, C. D., i Gadea, M. P., Sanchis, A., and Toselli, A. H. (2008). iatros: A speech and handwritting recognition system. In *V Jornadas en Tecnologías del Habla (VJTH'2008)*, pages 75–78.

Martín-Albo, D., Plamondon, R., and Vidal, E. (2014). Training of on-line handwriting text recognizers with synthetic text generated using the kinematic theory of rapid human movements. In *Proceedings of 14th International Conference on Frontiers in Handwriting Recognition*, pages 543–548. B.

Shivram, A., Ramaiah, C., Setlur, S., and Govindaraju, V. (2013). Ibm_ub_1: A dual mode unconstrained english handwriting dataset. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 13–17.

Toselli, A. H., i Gadea, M. P., and Vidal, E. (2007). On-line handwriting recognition system for tamil handwritten characters. In *3rd Iberian Conference on Pattern Recognition and Image Analysis*, pages 370–377. Springer-Verlag.

Attachment A

Task MT Quality and PE

.

The Impact of Machine Translation Quality on Human Post-editing

Philipp Koehn[◇⋆]
pkoehn@inf.ed.ac.uk
[◇]Center for Speech and Language Processing
The Johns Hopkins University

Ulrich Germann[⋆]
ugermann@inf.ed.ac.uk
[⋆]School of Informatics
University of Edinburgh

Abstract

We investigate the effect of four different competitive machine translation systems on post-editor productivity and behaviour. The study involves four volunteers post-editing automatic translations of news stories from English to German. We see significant difference in productivity due to the systems (about 20%), and even bigger variance between post-editors.

1 Introduction

Statistical machine translation (SMT) has made considerable progress over the past two decades. Numerous recent studies have shown productivity increases with post-editing of MT output over traditional work practices in human translation (e.g., Guerberof, 2009; Plitt and Masselot, 2010; Garcia, 2011; Pouliquen et al., 2011; Skadiņš et al., 2011; den Bogaert and Sutter, 2013; Vazquez et al., 2013; Green et al., 2013; Läubli et al., 2013).

The advances in statistical machine translation over the past years have been driven to a large extent by frequent (friendly) competitive MT evaluation campaigns, such as the shared tasks at the ACL WMT workshop series (Bojar et al., 2013) and IWSLT (Cettolo et al., 2013), and the NIST Open MT Evaluation.[1] These evaluations usually apply a mix of automatic evaluation metrics, most prominently the BLEU score (Papineni et al., 2001), and more subjective human evaluation criteria such as correctness, accuracy, and fluency.

How the quality increases measured by automatic metrics and subjective evaluation criteria relate to actual increases in the productivity of post-editors is still an open research question. It is also not clear yet if some machine translation approaches — say, syntax-based models — are better suited for post-editing than others. These relationships may very well also depend on the language pair in question and the coarse level of MT quality, from barely good enough for post-editing to almost perfect.

The pilot study presented in this paper investigates the influence of the underlying SMT system on post-editing effort and efficiency. The study focuses on translation of general news text from English into German, with translations created by non-professional post-editors working on output from four different translation systems. The data generated by this study is available for download.[2]

We find that the better systems lead to a productivity gain of roughly 20% and carry out in-depth analysis of editing behavior. A significant finding is the high variance in work styles between the different post-editors, compared to the impact of machine translation systems.

2 Related Work

Koponen (2012) examined the relationship between human assessment of post-editing efforts and objective measures such as post-editing time and number of edit operations. She found that segments that require a lot of reordering are perceived as being more difficult, and that long sentences are considered harder, even if only few words changed. She also reports larger variance between translators in post-editing *time* than in post-editing *operations* — a finding that we confirm here as well.

From a detailed analysis of the types of edits performed in sentences with long versus short post-edit times, Koponen et al. (2012) conclude that the observed differences in edit times can be explained at least in part also by the types of necessary edits and the associated cognitive effort. Deleting superfluous function words, for example, appears to be cognitively simple and takes little time, whereas inserting translations for untranslated words requires more cognitive effort

[1] http://www.nist.gov/itl/iad/mig/openmt.cfm

[2] http://www.casmacat.eu/index.php?n=Main.Downloads

Workshop on Humans and Computer-assisted Translation, pages 38–46,
Gothenburg, Sweden, 26 April 2014.

Table 1: News stories used in the study (size is given in number of sentences)

Source	Size	Title
BBC	49	Norway's rakfisk: Is this the world's smelliest fish?
BBC	47	Mexico's Enrique Pena Nieto faces tough start
CNN	45	Bradley Manning didn't complain about mistreatment, prosecutors contend
CNN	63	My Mexican-American identity crisis
Economist	55	Old battles, new Middle East
Guardian	38	Cigarette plain packaging laws come into force in Australia
NY Times	61	In a Constantly Plugged-In World, It's Not All Bad to Be Bored
NY Times	47	In Colorado, No Playbook for New Marijuana Law
Telegraph	95	Petronella Wyatt: I was bullied out of Oxford for being a Tory

and takes longer. They also compare post-editing styles of different post-editors working on identical post-editing tasks.

Another study by Koponen (2013) showed that inter-translator variance is lower in a controlled language setting when translators are given the choice of output from three different machine translation systems.

In the realm of machine translation research, there has been an increasing interest in the use of MT technology by post-editors. A major push are the two EU-funded research projects MATECAT[3] and CASMACAT[4], which are developing an open source translation and post-editing workbench (Federico et al., 2012; Alabau et al., 2013).

At this point, we are not aware of any study that compares directly the impact of different machine translation systems on post-editor productivity and behaviour.

3 Experimental Design

We thus carried out an experiment on an English–German news translation task, using output from four different SMT systems, post-edited by fluent bilingual native speakers of German with no prior experience in professional translation.

3.1 The Translation Task

The Workshop on Statistical Machine Translation (Bojar et al., 2013) organises an annual evaluation campaign for machine translation systems. The subject matter is translation of news stories from sources such as the New York Times or the BBC. We decided to use output from systems submitted to this evaluation campaign, not only because their output is freely available,[5] but also because it comes with automatic metric scores and human judgements of the translation quality.

The translation direction we chose was English–German, partly due to convenience (the authors of this study are fluent in both languages), but also because this language pair poses special challenges to current machine translation technology, due to the syntactic divergence of the two languages.

We selected data from the most recent evaluation campaign. The subset chosen for our post-editing task comprises 9 different news stories, originally written in English, with a total of 500 sentences. Details are shown in Table 1.

3.2 Machine Translation Systems

A total of 15 different machine translation systems participated in the evaluation campaign. We selected four different systems that differ in their architecture and use of training data:

- an anonymized popular online translation system built by a large Internet company (ONLINE-B)
- the syntax-based translation system of the University of Edinburgh (UEDIN-SYNTAX; Nadejde et al., 2013)
- the phrase-based translation system of the University of Edinburgh (UEDIN-PHRASE; Durrani et al., 2013)
- the machine translation system of the University of Uppsala (UU; Stymne et al., 2013)

In the 2013 WMT evaluation campaign, the systems translated a total of 3000 sentences, and their

[3] http://www.matecat.com/
[4] http://www.casmacat.eu/
[5] http://www.statmt.org/wmt13/results.html

Table 2: Machine translation systems used in the study, with quality scores in the WMT 2013 evaluation campaign.

System	BLEU	SUBJECTIVE
ONLINE-B	20.7	0.637
UEDIN-SYNTAX	19.4	0.614
UEDIN-PHRASE	20.1	0.571
UU	16.1	0.361

output was judged with the BLEU score against a professional reference translation and by subjective ranking. The scores obtained for the different systems on the full test set are shown in Table 2. The first three systems are fairly close in quality (although the differences in subjective human judgement scores are statistically significant), whereas the fourth system (UU) clearly lags behind. The best system ONLINE-B was ranked first according to human judgement and thus can be considered state of the art.

From casual observation, the syntax-based system UEDIN-SYNTAX succeeds more frequently in producing grammatically correct translations. The phrase-based system UEDIN-PHRASE, even though trained on the same parallel data, has higher coverage since it does not have the requirement that translation rules have to match syntactic constituents in the target language, which we presume is the main cause behind the lower BLEU score. The two systems use the same language model.

System UU is also a phrase based system, with a decoder that is able to consider the document level context. It was trained on smaller corpora for both the translation model and the language model.

We do not have any insight into the system ONLINE-B, but we conjecture that it is a phrase-based system with syntactic pre-reordering trained on much larger data sets, but not optimised towards the news domain.

Notice the inconsistency between BLEU score and subjective score for the two systems from the University of Edinburgh. Results from other evaluations have also shown (Callison-Burch et al., 2012) that current automatic evaluation metrics do not as much as human judges appreciate the strengths of the syntax-based system, which builds syntactic structures in the target language during translation. Hence, we were particularly interested how the syntax-based system fares with post-editors.

As mentioned above, the nine documents chosen for the post-editing task analysed in this paper (cf. Table 1) were part of the WMT 2013 evaluation data set. All nine documents had English as the original source language.

3.3 Post-Editors

We recruited four English-German bilingual, native German post-editors. Three were students, staff, or faculty at the University of Edinburgh; the fourth had been previously employed on a contractual basis for linguistic annotation work.[6] The post-editors had no professional experience with translation, and differed in language skills.

3.4 Assignment of MT Output

The goal of this study was to investigate how post-editors' behaviour and productivity are influenced by the quality of the underlying machine translation system. Ideally, we would want to present output from different systems to the same post-editor and see how their observable behaviour changes.

However, a post-editor who has seen the output from one MT system for a sentence will be at an advantage when post-editing the output from a second system, by having already spent significant time understanding the source sentence and considering the best translation choices.

Hence we used 4 different post-editors, each to post-edit the output in equal amounts from each of the 4 machine translation systems under investigation, so that each post-editor worked on each sentence once and the entire output from all systems was post-edited once by one of the 4 post-editors.

A concern in this setup is that we never know if we measure differences in post-editors or differences in machine translations systems when comparing the behaviour for any given sentence.

Therefore, each post-editor was assigned a translation for each sentence randomly from any of the machine translation systems. This random assignment allows us to marginalise out the dependence on the post-editor when assessing statistics for the different systems.

[6] The ordering here does not reflect the order of post-editors in the discussion later in this paper.

Table 3: Post-editing speed by editor and system.

System	seconds / word					words / hour				
	1	2	3	4	**mean**	1	2	3	4	**mean**
ONLINE-B	2.95	4.69	9.16	4.98	**5.46**	1,220	768	393	723	**659**
UEDIN-PHRASE	3.04	5.01	9.22	4.70	**5.45**	1,184	719	390	766	**661**
UEDIN-SYNTAX	3.03	4.41	9.20	4.97	**5.38**	1,188	816	391	724	**669**
UU	3.11	5.01	11.59	5.58	**6.35**	1,158	719	311	645	**567**
mean per editor	**3.03**	**4.78**	**9.79**	**5.05**		**1,188**	**753**	**368**	**713**	

4 Productivity

The primary argument for post-editing machine translation output as opposed to more traditional approaches is the potential gain in productivity. If translation professionals can work faster with machine translation, then this has real economic benefits. There are also other considerations, for example that post-editing might be done by professionals that are less skilled in the source language (Koehn, 2010).

We measure productivity by time spent on each sentence. This is not a perfect measure. When working on a news story, post-editors tend to speed up when moving down the story since they have already solved some reoccurring translation problems and get more familiar with the context.

4.1 Productivity by MT System

Our main interests is the average translation speed, broken down by machine translation system. The columns labelled "mean" in Table 3 show the results. While the differences are not big for the top three systems, the syntax-based system comes out on top.

We used bootstrap resampling to test the speed differences for statistical significance. Only system UU is significantly worse than the others (at p-level < 0.01), with about 20% lower productivity.

4.2 Productivity by Post-Editor

Post-editing speed is very strongly influenced by the post-editor's skill and effort. Our post-editors were very diverse, showing large differences in translation speed. See the columns labelled 1 to 4 in Table 3 for details.

In particular, post-editor 3 took more than three times as much time as the fastest (PE 1). According to a post-study interview with Post-Editor 3, there were two reasons for this. First, the post-editor was feeling a bit "under the weather" during the study and found it hard to focus. Second, (s)he found the texts very difficult to translate and struggled with idiomatic expressions and cultural references that (s)he did not understand immediately.

4.3 Productivity by System and Post-Editor

While the large differences between the post-editors are unfortunate when the goal is consistency in results, they provide some data on how post-editors of different skill levels are influenced by the quality of the machine translation systems.

Table 3 breaks down translation speed by machine translation system and post-editor. Interestingly, machine translation quality has hardly any effect on the fast Post-Editor 1, and the lower MT performance of system UU affects only Post-Editors 3 and 4. Post-Editor 2 is noticeably faster with UEDIN-SYNTAX — an effect that cannot be observed for the other post-editors. The differences between the other systems are not large for any of the post-editors.

Statistically significant — as determined by bootstrap resampling — are only the differences in post-editing speed for Post-Editor 3 with system UU versus ONLINE-B and UEDIN-PHRASE at p-level < 0.01, and against UEDIN-SYNTAX at p-level <0.02, and for Post-Editor 4 for UU versus UEDIN-PHRASE at p-level < 0.05. Note that the absence of statistical significance in our data has much to do with the small sample size; more extensive experiments may be necessary to ensure more solid findings.

5 Translation Edit Rate

Given the inherent difficulties in obtaining timing information, we can also measure the impact of machine translation system quality on post-editing effort in terms of how much the post-editors change the machine translation output, as done, for example in Cettolo et al. (2013).

Table 4: Edit rate and types of edits per system

System	HTER	ins	del	sub	shift	wide shift
ONLINE-B	35.7	4.8	7.4	18.9	4.6	5.8
UEDIN-PHRASE	37.9	5.5	7.4	20.0	5.0	6.6
UEDIN-SYNTAX	36.7	4.7	7.6	19.8	4.6	5.7
UU	43.7	4.6	11.4	21.9	5.8	7.2

Table 5: Edit rate and types of edits per post-editor

P-E	HTER	ins	del	sub	shift	wide shift
1	35.2	5.4	6.7	18.7	4.4	5.3
2	43.1	4.1	10.4	23.1	5.4	6.9
3	37.7	5.9	7.9	18.8	5.0	6.6
4	37.5	4.3	8.5	19.6	5.1	6.4

There are two ways to measure how much the machine translation output was edited by the post-editor. One way is to compare the final translation with the original machine translation output. This is what we will do in this section. In Section 6, we will consider which parts of the final translation were actually changed by the post-editor and discuss the difference.

5.1 HTER as Quality Measure

The edit distance between machine translation output and human reference translation can be measured in the number of insertions, deletions, substitutions and (phrasal) moves. A metric that simply counts the minimal number of such edit operations and divides it by the length of the human reference translation is the *translation edit rate*, short TER (Snover et al., 2006).

If the human reference translation is created from the machine translation output to minimise the number of edit operations needed for an acceptable translation, this variant is called *human-mediated* TER, or HTER. Note that in our experiment the post-editors are not strictly trying to minimise the number of edit operations — they may be inclined to make additional changes due to arbitrary considerations of style or perform edits that are faster rather than minimise the number of operations (e.g., deleting whole passages and rewriting them).

5.2 Edits by MT System

Table 4 shows the HTER scores — keep in mind our desiderata above — for the four systems. The scores are similar to the productivity number, with the three leading systems close together and the trailing system UU well behind.

Notably, we draw more statistically significant distinctions here. While as above, UU is significantly worse than all other systems (p-level $<$ 0.01), we also find that ONLINE-B is better than UEDIN-PHRASE (p-level $<$ 0.01).

Hence, HTER is a more sensitive metric than translation speed. This may be due to the fact that the time measurements are noisier than the count of edit operations. But it may also because HTER and productivity (i.e., time) do not measure the exactly the same thing. For instance, edits that require only a few keystrokes may be cognitively demanding (e.g., terminological choices), and thus take more time.

We cannot make any strong claim based on our numbers, but it is worth pointing out that post-editing UEDIN-SYNTAX was slightly faster than ONLINE-B (by 0.08 seconds/word), while the HTER score is lower (by 1 point). A closer look at the edit operations reveals that the post-edit of UEDIN-SYNTAX output required slightly fewer short and long shifts (movements of phrases), but more substitutions. Intuitively, moving a phrase around is a more time-consuming task than replacing a word. The benefit of a syntax-based system that aims to produce correct syntactic structure (including word order), may have real benefits in terms of post-editing time.

5.3 Edits by Post-Editor

Table 5 displays the edit rate broken down by post-editor. There is little correlation between edit rate and post-editor speed. While the fastest Post-Editor 1 produces translations with the smallest edit rate, the difference to two of the others (included the slowest Post-Editor 3) is not large. The

sec	current_translation
0.000	Norwegen ist es nicht oft auf die globale Agenda - und die meisten Nachrichten, wie es scheint.
0.508	Norwegen ist esnicht oft auf die globale Agenda - und die meisten Nachrichten, wie es scheint.
0.733	Norwegen ist enicht oft auf die globale Agenda - und die meisten Nachrichten, wie es scheint.
0.970	Norwegen ist enicht oft auf die globale Agenda - und die meisten Nachrichten, wie es scheint.
0.975	Norwegen ist nicht oft auf die globale Agenda - und die meisten Nachrichten, wie es scheint.
3.317	Norwegen ist nicht oft auf di globale Agenda - und die meisten Nachrichten, wie es scheint.
3.413	Norwegen ist nicht oft auf d globale Agenda - und die meisten Nachrichten, wie es scheint.
3.524	Norwegen ist nicht oft auf de globale Agenda - und die meisten Nachrichten, wie es scheint.
3.588	Norwegen ist nicht oft auf der globale Agenda - und die meisten Nachrichten, wie es scheint.
5.116	Norwegen ist nicht oft auf der globalen Agenda - und die meisten Nachrichten, wie es scheint.
17.986	Norwegen ist nicht oft auf der globalen Agenda - und die meisten , wie es scheint.
19.386	Norwegen ist nicht oft auf der globalen NachrichtenAgenda - und die meisten , wie es scheint.
20.116	Norwegen ist nicht oft auf der globalen Nachrichtengenda - und die meisten , wie es scheint.
20.196	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten , wie es scheint.
20.298	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten , wie es scheint.
29.596	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheint.
31.459	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten schein.
31.659	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheine.
31.796	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen.
32.060	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen .
34.283	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen d.
34.380	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen da.
34.443	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen das.
34.636	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen das .
35.507	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen das s.
35.637	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen das so.
35.744	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen das so .
35.949	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen das so z.
36.053	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen das so zu.
36.166	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen das so zu .
36.805	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen das so zu w.
38.011	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen das so zu wo.
38.394	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen das so zu wol.
38.699	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen das so zu woll.
38.795	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen das so zu wolle.
38.947	Norwegen ist nicht oft auf der globalen Nachrichtenagenda - und die meisten scheinen das so zu wollen.
char	mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmttmmmmmmmmtmpppppppppppptmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmttttttttttttttttttttttm
word	m m m m m x x x m m m m x t t t t m

Figure 1: Analysis of the post-editing process: Most editing actions that result in changes in the translation are adding or deleting of a character by a keystroke. However, notice the cut (17.986 sec) and paste (19.386 sec) action. Each character can be traced either to the original machine translation output (m), a typing action of the post-editor (t), or a pasting action of the post-editor (p). This then allows tokens in the output classified as either original MT (m), typed (t), pasted (not in figure) or partially edited (x).

outlier here is Post-Editor 2, whose output has a much larger edit rate.

6 Editing Actions

The HTER is an analysis of the *product* of post-editing. The final translation is compared to the original machine translation output. In this section, we examine how the *process* of post-editing is influenced by the machine translation system.

Our post-editing workbench provides detailed logging of each HCI interaction (key strokes, mouse clicks, etc.). This allows us to reconstruct how a translation was generated. See Figure 1 for an example how a translated was edited.

Table 6: Character provenance by system

System	MT	typed	pasted
ONLINE-B	68.3	28.0	3.3
UEDIN-PHRASE	62.9	31.3	5.2
UEDIN-SYNTAX	65.9	29.1	4.5
UU	56.1	37.9	5.6

6.1 Character Provenance by System

If we follow the editing actions, we can trace the origin of each letter in the final output: was it part of the original MT output, was it typed in by the user, or moved in a cut and paste action? Table 6 breaks down the characters in the final translations

Table 7: Token provenance by system

System	MT	typed	pasted	edited
ONLINE-B	65.2	21.4	2.3	10.8
UEDIN-PHRASE	60.5	24.7	3.9	10.6
UEDIN-SYNTAX	62.6	22.4	3.4	11.3
UU	53.2	31.0	4.0	11.7

by origin for each system. The numbers correspond to the HTER scores, with a remarkable consistency ranking for typed and pasted characters.

6.2 Token Provenance by System

We perform a similar analysis on the word level, introducing a fourth type of provenance: words whose characters are of mixed origin, i.e., words that were partially edited. Table 7 shows the numbers for each machine translation system. The suspicion from the HTER score that the syntax-based system UEDIN-SYNTAX requires less movement is not confirmed by these numbers. There are significantly more words moved by pasting (3.4%) than for ONLINE-B (2.3%). In general, cutting and pasting is not as common as the HTER score would suggest: the two types of shifts moved 10.3% and 10.2% of phrases, respectively. It seems that most words that could be moved are rather deleted and typed again.

6.3 Behaviour By Post-Editor

The post-editors differ significantly in their behaviour, as the numbers in Table 8 illustrate. Post-Editor 1, who is the fastest, leaves the most characters unchanged (72.9% vs. 57.7–64.4% for the others). Remarkably, this did not result in a dramatically lower HTER score (recall: 35.2 vs. 37.5–43.1 for the others).

Post-Editor 3, while taking the longest time, does not change the most number of characters. However, (s)he uses dramatically more cutting and pasting. Is this activity particularly slow? One way to check is to examine more closely how the post-editors spread out their actions over time.

Table 8: Character provenance by post-editor

Post-Editor	MT	typed	pasted
1	72.9	22.9	3.5
2	57.7	39.4	2.7
3	58.9	29.5	10.7
4	64.4	33.5	1.9

7 Editing Activities

Koehn (2009) suggests to divide up the time spent by translators and post-editors into intervals of the following types:

- initial pauses: the pause at the beginning of the translation, if it exists
- end pause: the pause at the end of the translation, if it exists
- short pause of length 2–6 seconds
- medium pauses of length 6–60 seconds
- big pauses longer than 60 seconds
- various working activities (in our case just typing and mouse actions)

When we break up the time spent on each activity and normalise it by the number of words in the original machine translation output, we get the numbers in Table 9, per machine translation system and post-editor.

The worse quality of the UU system causes mainly more work activity, big medium pauses. Each contributes roughly 0.3 seconds per word. The syntax-based system UEDIN-SYNTAX may pose fewer hard translation problems (showing up in initial and big pauses) than the HTER-preferred ONLINE-B system, but the effect is not strong.

We noted that ONLINE-B has a statistically significant better HTER score than UEDIN-PHRASE. While this is reflected in the additional working activity for the latter (2.41 sec./word vs. 2.26 sec./word), time is made up in the pauses. Our data is not sufficiently conclusive to gain any deeper insight here — it is certainly a question that we want to explore in the future.

The difference in post-editors mirrors some of the earlier findings: The number of characters and words changed leads to longer working activity, but the slow Post-Editor 3 is mainly slowed down by initial, big and medium pauses, indicating difficulties with solving translation problems, and not slow cutting and pasting actions. The faster Post-Editor 1 rarely pauses long and is quick with typing and mouse movements.

8 Conclusion

We compared how four different machine translation systems affect post-editing productivity and behaviour by analysing final translations and user

Table 9: Time spent on different activities, by machine translation system (top) and post-editor (bottom).

System	initial pause	big pause	med. pause	short pause	end pause	working
ONLINE-B	0.37 s/w	0.61 s/w	1.88 s/w	0.30 s/w	0.00 s/w	2.26 s/w
UEDIN-PHRASE	0.32 s/w	0.55 s/w	1.74 s/w	0.32 s/w	0.00 s/w	2.41 s/w
UEDIN-SYNTAX	0.32 s/w	0.50 s/w	1.90 s/w	0.31 s/w	0.00 s/w	2.30 s/w
UU	0.28 s/w	0.74 s/w	2.14 s/w	0.34 s/w	0.00 s/w	2.75 s/w

Post-Editor	initial pause	big pause	med. pause	short pause	end pause	working
1	0.35 s/w	0.01 s/w	0.63 s/w	0.27 s/w	0.00 s/w	1.76 s/w
2	0.04 s/w	0.19 s/w	1.13 s/w	0.35 s/w	0.00 s/w	3.06 s/w
3	0.91 s/w	1.85 s/w	3.99 s/w	0.29 s/w	0.00 s/w	2.53 s/w
4	0.02 s/w	0.36 s/w	1.94 s/w	0.35 s/w	0.00 s/w	2.33 s/w

activity data. The best system under consideration yielded abut 20% better productivity than the worst, although the three systems on top are not statistically significantly different in terms of productivity.

We noted differences in metrics that measure productivity and edit distance metrics. The latter allowed us to draw more statistically significant conclusions, but may measure something distinct. Productivity is the main concern of commercial use of post-editing machine translation, and we find that better machine translation leads to less time spent on editing, but more importantly, less time spent of figuring out harder translation problems (indicated by pauses of more than six seconds).

Finally, an important finding is that the differences between post-editors is much larger than the difference between machine translation systems. This points towards the importance of skilled post-editors, but this finding should be validated with professional post-editors, and not the volunteers used in this study.

Acknowledgements

This work was supported under the CASMACAT project (grant agreement N^{o} 287576) by the European Union 7th Framework Programme (FP7/2007-2013).

References

Alabau, Vicent, Ragnar Bonk, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Jesús González, Philipp Koehn, Luis Leiva, Bartolomé Mesa-Lao, Daniel Ortiz, Herve Saint-Amand, Germán Sanchis, and Chara Tsoukala. 2013. "CASMACAT: An open source workbench for advanced computer aided translation." *The Prague Bulletin of Mathematical Linguistics*, 100:101–112.

Bojar, Ondřej, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. "Findings of the 2013 Workshop on Statistical Machine Translation." *Proceedings of the Eighth Workshop on Statistical Machine Translation*, 1–44. Sofia, Bulgaria.

Callison-Burch, Chris, 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*, 10–48. Montreal, Canada.

Cettolo, Mauro, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2013. "Report on the 10th IWSLT evaluation campaign." *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*.

den Bogaert, Joachim Van and Nathalie De Sutter. 2013. "Productivity or quality? Let's do both." *Machine Translation Summit XIV*, 381–390.

Durrani, Nadir, Barry Haddow, Kenneth Heafield, and Philipp Koehn. 2013. "Edinburgh's machine translation systems for European language pairs." *Proceedings of the Eighth Workshop on Statistical Machine Translation*, 114–121. Sofia, Bulgaria.

Federico, Marcello, Alessandro Cattelan, and Marco Trombetti. 2012. "Measuring user productivity in machine translation enhanced computer assisted translation." *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA)*.

Garcia, Ignacio. 2011. "Translating by post-editing: is it the way forward?" *Machine Translation*, 25(3):217–237.

Green, Spence, Jeffrey Heer, and Christopher D. Manning. 2013. "The efficacy of human post-editing for language translation." *ACM Human Factors in Computing Systems (CHI)*.

Guerberof, Ana. 2009. "Productivity and quality in mt post-editing." *MT Summit Workshop on New Tools for Translators*.

Koehn, Philipp. 2009. "A process study of computer-aided translation." *Machine Translation*, 23(4):241–263.

Koehn, Philipp. 2010. "Enabling monolingual translators: Post-editing vs. options." *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 537–545. Los Angeles, California.

Koponen, Maarit. 2012. "Comparing human perceptions of post-editing effort with post-editing operations." *Pro-*

ceedings of the Seventh Workshop on Statistical Machine Translation, 227–236. Montreal, Canada.

Koponen, Maarit. 2013. "This translation is not too bad: an analysis of post-editor choices in a machine-translation post-editing task." *Proceedings of Workshop on Post-editing Technology and Practice*, 1–9.

Koponen, Maarit, Wilker Aziz, Luciana Ramos, and Lucia Specia. 2012. "Post-editing time as a measure of cognitive effort ." *AMTA 2012 Workshop on Post-Editing Technology and Practice (WPTP 2012)*, 11–20. San Diego, USA.

Läubli, Samuel, Mark Fishel, Gary Massey, Maureen Ehrensberger-Dow, and Martin Volk. 2013. "Assessing post-editing efficiency in a realistic translation environment." *Proceedings of Workshop on Post-editing Technology and Practice*, 83–91.

Nadejde, Maria, Philip Williams, and Philipp Koehn. 2013. "Edinburgh's syntax-based machine translation systems." *Proceedings of the Eighth Workshop on Statistical Machine Translation*, 170–176. Sofia, Bulgaria.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. *BLEU: a Method for Automatic Evaluation of Machine Translation*. Tech. Rep. RC22176(W0109-022), IBM Research Report.

Plitt, Mirko and Francois Masselot. 2010. "A productivity test of statistical machine translation post-editing in a typical localisation context." *Prague Bulletin of Mathematical Linguistics*, 93:7–16.

Pouliquen, Bruno, Christophe Mazenc, and Aldo Iorio. 2011. "Tapta: A user-driven translation system for patent documents based on domain-aware statistical machine translation." *Proceedings of th 15th International Conference of the European Association for Machine Translation (EAMT)*, 5–12.

Skadiņš, Raivis, Maris Puriņš, Inguna Skadiņa, and Andrejs Vasiļjevs. 2011. "Evaluation of SMT in localization to under-resourced inflected language." *Proceedings of the 15th International Conference of the European Association for Machine Translation (EAMT)*, 35–40.

Snover, Matthew, Bonnie J. Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. "A study of translation edit rate with targeted human annotation." *5th Conference of the Association for Machine Translation in the Americas (AMTA)*. Boston, Massachusetts.

Stymne, Sara, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2013. "Tunable distortion limits and corpus cleaning for SMT." *Proceedings of the Eighth Workshop on Statistical Machine Translation*, 225–231. Sofia, Bulgaria.

Vazquez, Lucia Morado, Silvia Rodriguez Vazquez, and Pierrette Bouillon. 2013. "Comparing forum data post-editing performance using translation memory and machine translation output: A pilot study." *Machine Translation Summit XIV*, 249–256.