Improving the Multi-Modal Post-Editing (MMPE) CAT Environment based on Professional Translators' Feedback

Nico Herbig[1] nico.herbig@dfki.de
Santanu Pal[2] santanu.pal2@wipro.com
Tim Düwel[1] tim.duewel@dfki.de
Raksha Shenoy[1,3] raksha.shenoy@dfki.de
Antonio Krüger[1] antonio.krueger@dfki.de
Josef van Genabith[1,3] josef.vangenabith@uni-saarland.de

[1]German Research Center for Artificial Intelligence (DFKI), Saarland Informatics Campus, Germany

[2]Wipro AI, Bangalore, India

[3]Department of Language Science and Technology, Saarland University, Germany

Abstract

More and more professional translators are switching to the use of post-editing (PE) to increase productivity and reduce errors. Even though PE requires significantly less text production, current computer-aided translation (CAT) interfaces still heavily focus on traditional mouse and keyboard input and ignore other interaction modalities to support PE operations. Recently, a multi-modal post-editing (MMPE) CAT environment combining pen, touch, speech, multi-modal interaction, and conventional mouse and keyboard input possibilities was presented. The design and development of the first version of MMPE were informed by extensive consultation with professional translators. This paper details how MMPE was since further refined by professional translators' judgments on which interaction modalities are most suitable for which PE task and more general qualitative findings. In particular, we describe how the layout was adapted to avoid confusion, which visualization aids were added, how the handwriting input was improved, and how we enabled multi-word reordering through touch drag and drop, as well as how the speech and multi-modal interaction components were enhanced. Finally, we present a sneak preview of how we integrated eye tracking not only for logging but also as an input modality that can be used in combination with keyboard or speech commands.

1 Introduction

Due to significant improvements in machine translation (MT) quality over the past years,[1] more and more professional translators are integrating this technology into their translation workflows (Zaretskaya et al., 2016; Zaretskaya and Seghiri, 2018). The process of using a pre-translated but potentially erroneous text, often MT output, and improving it to create the final translation is called post-editing (PE). Translators' perceptions regarding post-editing range

[1]WMT translation task: `http://matrix.statmt.org/matrix` for newstest2019, accessed 23/07/2020

from strong dislikes in older research (Lagoudaki, 2009; Wallis, 2006), to questioning its benefit and being cautious about PE (Gaspari et al., 2014; Koponen, 2012), to seeing it as a threat to their profession (Moorkens, 2018). However, many users have dated perceptions of MT and often prefer PE over translating from scratch when confronted with modern MT (Green et al., 2013). Furthermore, perceptions regarding PE of experienced translators appear more negative (Moorkens and O'Brien, 2015) than those of novice translators (Yamada, 2015).

Even though translators remain critical regarding PE, productivity gains of 36% when using modern neural MT for PE (Toral et al., 2018) have been shown. Since the task changes from mostly text production to comparing and adapting MT and translation memory (TM) proposals, a thorough re-investigation of interface designs is required. For this, Herbig et al. (2019a) conducted an elicitation study to explore which interaction modalities could well support the PE process and found that translators envision PE interfaces relying on touch, pen, and speech input combined with mouse and keyboard as particularly useful. Initial tests by Teixeira et al. (2019) using touch for reordering and speech dictation further showed promising practical results. The recently presented MMPE prototype (Herbig et al., 2020c) combines even more modalities, namely standard mouse & keyboard input with touch, pen, and speech interactions for PE of MT, and their study shows that translators are enthusiastic about having these interaction possibilities. For deletions and reorderings, pen and touch modalities received good feedback and were efficient, while for insertions and replacements, speech and multi-modal input were perceived positively.

Apart from these main results, Herbig et al. (2020b) also presented a variety of qualitative findings. Here, we leverage this feedback to extend and improve the original MMPE CAT environment through a variety of layout changes, enhanced interaction flexibility, and others, as discussed below. Furthermore, we present a first glimpse of how we include eye tracking combined with speech or keyboard input as an additional interaction modality.

2 Related Work

This section presents related research on translation environments and pays special attention to interaction modalities other than mouse and keyboard.

2.1 CAT and Post-Editing

So-called CAT (computer-aided translation) environments offer features like MT and TM (translation memory) together with quality estimation and concordance functionality (Federico et al., 2014), alignments between source and MT (Schwartz et al., 2015), interactive MT offering assistance such as auto-completion (Green et al., 2014b,a), or intelligibility assessments (Coppers et al., 2018; Vandeghinste et al., 2016, 2019). Due to this feature-richness, most professional translators use CAT tools daily (van den Bergh et al., 2015).

Even though TM is still often valued higher than MT (Moorkens and O'Brien, 2017), in an experiment Vela et al. (2019) showed that professional translators choose MT in 80% of the cases when confronted with the choice between translation from scratch, TM, and MT, thus highlighting the importance of PE MT output. In terms of time savings, Zampieri and Vela (2014) find that PE is on average 28% faster for technical translations, Aranberri et al. (2014) show translation throughput is increased for both professionals and lay users when they do PE, and Läubli et al. (2013) find that PE also increases productivity in realistic environments. Apart from saving time, Green et al. (2013) showed that PE also reduces errors.

Obviously, the interaction pattern during PE changes in comparison to translation from scratch, leading to a significantly reduced amount of mouse and keyboard events (Carl et al., 2010; Green et al., 2013). Therefore, other modalities, in addition to mouse and keyboard, have been explored for PE, as we discuss in the next section.

2.2 Multi-Modal Approaches

Using automatic speech recognition has a long history for traditional translation from scratch (Dymetman et al., 1994; Brousseau et al., 1995), and dictating translations that are then manually transcribed by secretaries dates back even further (Theologitis, 1998). For PE, the more recent study of the SEECAT (Martinez et al., 2014) environment supporting automatic speech recognition (ASR) argues that its combination with typing could boost productivity. According to a survey by Mesa-Lao (2014), PE trainees have a positive attitude towards speech input and would consider adopting it, but only as a complement to other modalities. In a small-scale study, Zapata et al. (2017) found that ASR for PE was faster than ASR for translation from scratch. Nowadays, commercial CAT tools like memoQ and MateCat are also beginning to integrate ASR.

The CASMACAT tool (Alabau et al., 2013) further allows users to input text by handwriting with an e-pen in a separate area, where the handwriting is recognized and placed at the cursor position. A vision paper by Alabau and Casacuberta (2012) proposes to instead PE sentences with few errors by using copy-editing-like pen input directly on the text.

Studies on mobile PE via touch and speech within the Kanjingo app (O'Brien et al., 2014; Torres-Hostench et al., 2017) show that participants liked to drag and drop words for reordering. They further preferred speech input when translating from scratch, but used the iPhone keyboard for small modifications. Zapata (2016) also explores the use of speech and touch; however, the study did not focus on PE and used Microsoft Word instead of a proper CAT environment.

Teixeira et al. (2019) explore touch and speech in unaided translation, translation using TM, and translation using MT conditions. In their studies, touch input received poor feedback: Their visualization, having a tile per word that could be dragged and dropped, made reading more complicated, and touch insertions were rather complex in their implementation. In contrast, speech dictation was shown to be quite useful and even preferred to mouse and keyboard by half of the participants.

An elicitation study by Herbig et al. (2019a), exploring interaction modalities for PE in a structured way, indicates that pen, touch, and speech interaction should be combined with mouse and keyboard to improve PE of MT. Based on these findings, Herbig et al. (2020c,b) set out and built a prototype called MMPE. It combines mouse and keyboard input with pen, touch, and speech modalities: "Users can directly cross out or hand-write new text, drag and drop words for reordering, or use spoken commands to update the text in place" (Herbig et al., 2020c). Furthermore, their system focused on easily interpretable logs for translation process research that directly show whether users deleted, inserted, reordered, or replaced tokens during PE, thus improving on prior works' logs, which usually only show which keypresses occurred at which timestamps. Their study with professional translators suggested that "pen and touch interaction are suitable for deletion and reordering tasks, while they are of limited use for longer insertions. On the other hand, speech and multi-modal combinations of select & speech are considered suitable for replacements and insertions but offer less potential for deletion and reordering" (Herbig et al., 2020b).

In summary, previous research suggests that professional translators can increase productivity and reduce errors through PE; however, translators themselves are not always eager to switch to PE. It has been argued that the PE process might be better supported by using different modalities in addition to the conventional mouse and keyboard approaches, and a few prototypes showed promising results. The most diverse investigation of multi-modal interaction for PE by Herbig et al. (2020b) provided interesting insights into the modalities' advantages and disadvantages. However, the study also presented plenty of qualitative findings. Based on these, our work further improves and extends MMPE, as discussed in the next sections.

3 Improvements and Extensions to MMPE

This section presents the MMPE prototype and how we extended it. Overall, the prototype combines pen, touch, eye, and speech input with a traditional mouse and keyboard approach for PE of MT and focuses on professional translators in an office setting. We will introduce the full prototype and focus on how the original implementation (presented in Herbig et al. (2020c)) was extended and improved based on the feedback received from professional translators in Herbig et al. (2020b).

3.1 Hardware and Software Overview

The frontend is the key MMPE component to explore different interaction modalities, and was developed using Angular[2]. The backend, implemented using node.js[3], was kept rather minimal, only allowing saving and loading of projects from JSON files, and forwarding text/audio to spellchecking or audio transcription services. We continue to focus on the frontend, and only slightly extend the backend services where necessary.

To optimally support the developed interactions, Herbig et al. (2020c) proposed to use a large touch and pen screen that can be tilted (see Figure 1), specifically the Wacom Cintiq Pro 32 inch display with the Flex Arm. The tilting mechanism thereby aims to better support handwriting. While this setup is rather expensive (ca. 3500 Euro), the web-based application could be used on any cheaper touch and pen device, as the usage of Bootstrap[4] supports layouts on different resolutions. However, one should make sure that the display is large and reacts quickly to handwritten input.

Figure 1: Hardware setup with a large tiltable screen, a digital pen, mouse, keyboard, a headset, and an eye tracker.

Furthermore, a headset should be used for the prototype, as the transcription used for interpreting speech commands often yields bad results when using only internal microphones. The original study used a Sennheiser PC 8 here (ca. 35 Euro), but informal tests showed that most external microphones work well. A remote eye tracker (the Tobii 4C, ca. 200 Euro) is

[2] https://angular.io/, accessed 27/07/2020
[3] https://nodejs.org/en/, accessed 27/07/2020
[4] https://getbootstrap.com/, accessed 27/07/2020

attached to the screen and calibrated to the user with the provided eye tracking software, but other eye trackers could of course also be integrated. Naturally, MMPE also allows input using mouse and keyboard.

3.2 Overall Layout

Figure 2 shows our implemented horizontal source-target layout. Each segment's status (unedited, edited, confirmed) is visualized between source and target. On the far right, support tools are offered, as explained in Section 3.3. The top of the interface shows a toolbar where users can save, load, and navigate between projects, and enable or disable spellchecking, whitespace visualization, speech recognition, and eye tracking, which will be discussed below.

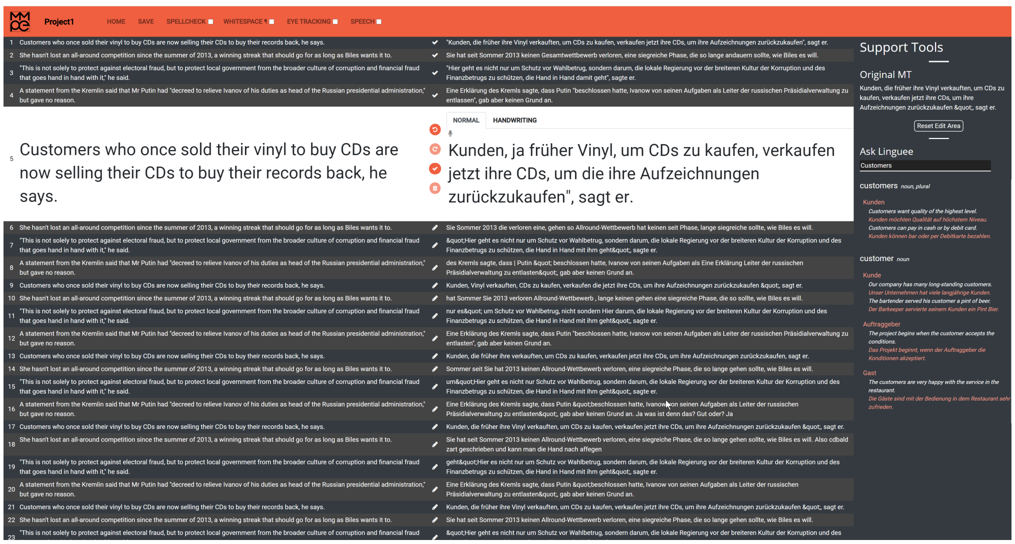


Figure 2: Screenshot of the interface.

The current segment is enlarged, thereby offering space for handwritten input and allowing users to view the context while still seeing the current segment in a comfortable manner (as requested in Herbig et al. (2019a)). The view for the current segment is further divided into the source segment (left) and tabbed editing planes for the target (right), one for handwriting and drawing gestures, and one for touch deletion & reordering, as well as standard mouse and keyboard input. By clicking on the tabs at the top, the user can quickly switch between the two modes. As the prototype focuses on PE, the target views initially show the MT proposal to be edited. The reason for having two editing fields instead of only one is that some interactions are overloaded, e.g., a touch drag can be interpreted as both hand-writing and reordering, depending on the tab in which it is performed. Undo and redo functionality and segment confirmation are also implemented: (1) By using hotkeys, (2) through buttons between source and target (next to an additional button for touch deletions, see below), or (3) through speech commands.

We decided to stick with the enlarged visualization of the current segment, as participants in Herbig et al. (2020b) liked the large font size. However, the original prototype had the two target views (handwriting and default editing) next to each other. Thus, overall, three neighboring views. This was perceived as unintuitive, leading to much confusion, especially at the beginning of the experiment, when participants did not remember which target view supported which features. Therefore, we combined the two, clearly labelling which mode does what, and allowing switching between them using the tabs quickly. The combination has the additional advantages that the interface becomes symmetric (as there is only one source and target for the previous, the current, and the remaining segments). Furthermore, the space for hand-writing increases even further, and the layout on smaller displays with insufficient space to nicely visualize three text boxes next to each other is improved.

Currently, we are adding further customization possibilities as requested by the participants in Herbig et al. (2020b), e.g., to adapt the font size or to switch between displaying source and target side by side or one above the other.

3.3 Support Tools and Visualization Aids

In the **support tools** (Figure 2 right), the user can reset the target text to the unedited MT output using a button. Furthermore, a bilingual concordancer is offered: When entering a word in the search box or clicking/touching a word in the source view on the left, the Linguee[5] website is queried to show the word in context and display its primary and alternative translations. Here, we are planning to integrate further features like Google direct answers.

Spellchecking (Figure 3a) can be enabled in the navigation bar at the top of the interface: For this, MMPE analyzes the target text using either the browser's integrated spellchecker, the node.js simple-spellchecker package[6], or the Microsoft Cognitive Services spellchecker. While Herbig et al. (2020b) did not investigate this feature in detail, we believe it to be essential for practical usage, since the unavailability of spellchecking was criticized in Teixeira et al. (2019).

A feature that we newly introduced into the prototype because it was requested in Herbig et al. (2020b) was the **visualization of whitespaces**, which can also be enabled in the navigation bar. Figure 3b shows the visualized spaces and line breaks, commonly known from, e.g., Microsoft Word.

(a) Spell checking. (b) Visualization of whitespaces.

Figure 3: Spellchecking and whitespace visualization.

3.4 Hand-writing View

Hand-writing in the hand-writing tab (see Figure 4) is recognized using the MyScript Interactive Ink SDK[7], which worked well in Herbig et al. (2020b). The input field further offers drawing gestures[8] like strike-through or scribble for deletions, breaking a word into two (draw a line from top to bottom), and joining words (draw a line from bottom to top). If there is a lack of space to hand-write the intended text, the user can create such space by breaking the line (draw a long line from top to bottom). All changes are immediately applied, e.g., deleting a word by strike-through removes it immediately and does not show it in a struck-through visualization. A small button at the top right of the editor can be used to convert handwritten text into computer font; however, this is only for cosmetic reasons and not required for the functionality. The editor further shows the recognized input immediately at the top of the drawing view in a small gray font. Clicking on a word here offers alternatives that can be easily selected (see Figure 4).

[5]`https://www.linguee.com/`, accessed 27/07/2020

[6]`https://www.npmjs.com/package/simple-spellchecker`, accessed 27/07/2020

[7]`https://developer.myscript.com/`, accessed 27/07/2020

[8]`https://developer.myscript.com/docs/concepts/editing-gestures/`, accessed 27/07/2020

Apart from using the pen, the user can use his/her finger or the mouse for hand-writing. However, most participants in Herbig et al. (2020b) preferred the pen to finger hand-writing for insertions and replacements due to its precision, although some considered it less direct than finger input. One participant in the study even used the strike-through deletion with the mouse; therefore, we decided to keep all three hand-writing options.

In Herbig et al. (2020b), participants highly valued deletion by strike-through or scribbling through the text, as this would nicely resemble standard copy-editing. However, hand-writing for replacements and insertions was considered to work well only for short modifications. For more extended changes, participants argued that one should instead fall back to typing or speech commands. An issue that might have influenced this finding was that in the evaluated version of MMPE, an unintended change of the currently selected segment happened when the palm of the hand touched another piece of text. As it is common to lay down one's hand while writing, we now prevent this unintended segment change by ignoring palm touches.

Furthermore, participants found the gesture to create space (drawing a vertical line at the position) often hard to accomplish, which we improved by increasing the lineheight. For the first line, we added even more space to the top, to prevent a drawing from starting in the text box containing the recognized text, where it would be ignored, thereby improving the user experience. Last, we deactivated the button of the digital pen, as it frequently resulted in unintended right-clicks triggering the context menu.

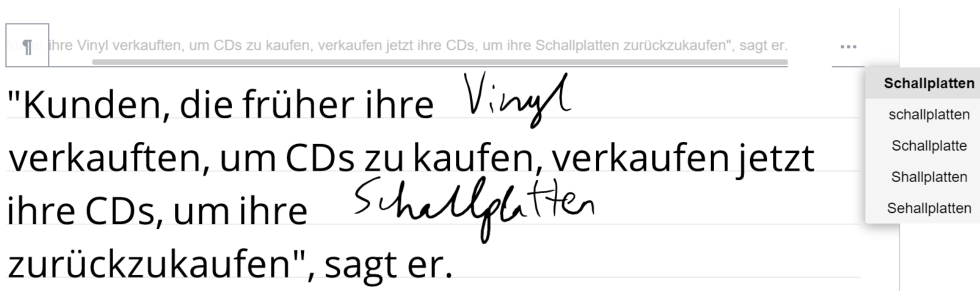


Figure 4: Hand-writing and alternatives after clicking the word "Schallplatten" in the recognized text at the top.

3.5 Default Editing View

Below we dive deeper into the interaction possibilities supported by the default editing view, and focus on improvements to these.

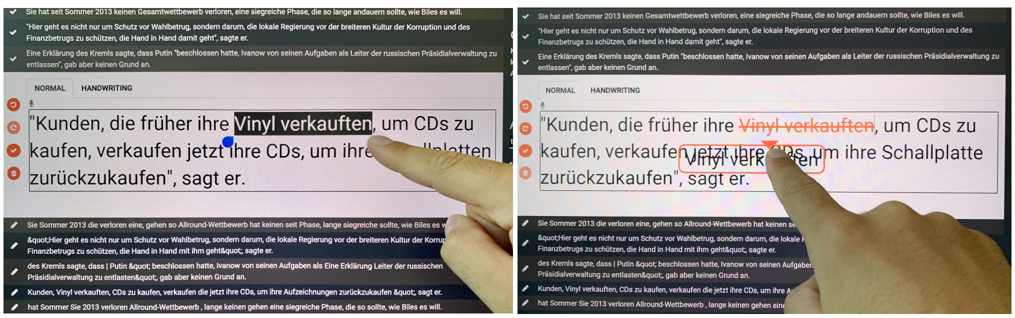
3.5.1 Mouse and Keyboard Input

Since mouse and keyboard inputs were still the fastest for insertion and replacement operations in the study by Herbig et al. (2020b), we naturally kept this feature, including the possibility of navigating using hotkeys like ctrl+arrow keys or copy & pasting using ctrl+c/v. Marking and drag and drop with the mouse are also supported as in any text processing application. However, since participants in the study stated that mouse and keyboard only work well due to years of experience and muscle memory, we will focus on other modalities in the remainder of this section.

3.5.2 Touch Reorder and Delete

Reordering using (pen or finger) touch is supported with a simple drag and drop procedure (see Figure 5b): For this, we visualize the picked-up word(s) below the touch position and show the calculated current drop position through a small arrow element. Spaces between words and punctuation marks are automatically fixed, i.e., double spaces at the pickup position are removed and missing spaces at the drop position are inserted. Furthermore, for the German language, nouns are automatically capitalized using the list of nouns from Wiktionary[9]. This reordering functionality is strongly related to Teixeira et al. (2019); however, only the currently dragged word is temporarily visualized as a tile to offer better readability.

Touch reordering was highlighted as particularly useful or even “perfect” as it “nicely resembles a standard correction task”, and received the highest subjective scores and lowest time required for reordering (Herbig et al., 2020b). Nevertheless, the old reorder only supported moving one word at a time, not whole sub-phrases or parts of words, which is naturally needed in actual PE settings. Now, users have two options: (1) They can drag and drop single words by starting a drag directly on top of a word, or (2) they can double-tap to start a selection process, define which part of the sentence should be selected (e.g., multiple words or a part of a word, see Figure 5a), and then move it (see Figure 5b).



(a) Multi-word selection using touch.

(b) Touch reorder.

Figure 5: Touch multi-word selection and reordering.

While this allows a much more flexible reorder functionality, it has the disadvantage that double-tap can no longer be used to delete words, as was the case in Herbig et al. (2020c). However, as strike-through in the hand-writing view was also highly liked for deletion, we think removing this functionality does not harm overall usability. Furthermore, we added a delete button alongside the undo/redo/confirm buttons so that users can still delete using touch by selecting text through double-tap and pressing the button then. Overall we believe that the increased flexibility should enhance usability, even though touch deletion of single words became slightly more complicated.

Several participants in Herbig et al. (2020b) noted that the text was jumping around when reordering a word from the end of a line: By immediately removing the picked-up word from the text, all remaining words moved to the front, and the placeholder element, was taking up space that also pushed words from line to line while dragging. We have now solved this issue by keeping the word(s) in the old position in a struck-through appearance (see Figure 5b), showing a copy of the word below the finger/pen, and only removing the actual word(s) on drop. Furthermore, the visualization was redesigned to make the drop position clearer without taking up any space, and highlighting the picked-up text better.

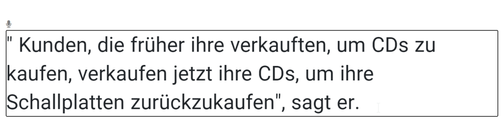
[9] `https://en.wiktionary.org/wiki/Category:German\_noun\_forms`, accessed 27/07/2020

3.6 Speech

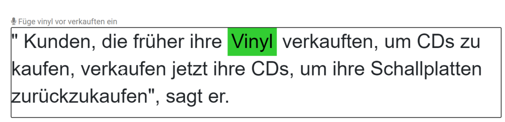
To minimize lag during speech recognition, we use a streaming approach, sending the recorded audio to IBM Watson servers to receive a transcription, which is then interpreted in a command-based fashion. Thus, the speech module not only handles dictations as in Teixeira et al. (2019) but can correct mistakes in place. The transcription itself is shown at the top of the default editing view next to a microphone symbol (see Figure 6). As commands, post-editors can "*insert*", "*delete*", "*replace*", and "*reorder*" words or sub-phrases. To specify the position if it is ambiguous, anchors can be specified, e.g., "*after*"/"*before*"/"*between*", or the occurrence of the token ("*first*"/"*second*"/"*last*") can be defined. A full example is "*replace A between B and C by D*", where A, B, C, and D can be words or sub-phrases. Due to recognition and practicality issues, character-level speech commands are not supported, so instead of deleting an ending, one should replace the word, e.g., "*replace finding with find*" instead of "*delete i n g from finding*". Again, spaces between words and punctuation marks are automatically fixed, and for German, nouns are capitalized as described for touch reordering.

In Herbig et al. (2020b), speech received good ratings for insertions and replacements, but worse ratings for reorderings and deletions. According to the participants, speech would become especially compelling for longer insertions and would be preferable when commands remain simple. However, it was considered problematic in shared offices and would be complex to formulate commands while mentally processing text. To limit the complexity of speech commands, we added further synonyms (e.g., "*write*" or "*put*" as alternatives to "*insert*") and allow users to specify anchors by occurrence (e.g., "*delete last A*"). Thus, we increase flexibility and offer more natural commands that participants had used in Herbig et al. (2020b), but which were not supported back then. Furthermore, we now allow modifying punctuation marks (e.g., "*delete comma after nevertheless*"), automatically capitalize words inserted at the beginning, uncapitalize them when reordered to other positions, capitalize the second word when deleting the first in the sentence, and so on. Furthermore, users can now choose to restate the whole sentence when MT quality is low, and in general, we allow dictations.

We improved user feedback regarding speech commands: On the one hand, invalid commands display why they are invalid below the transcription (e.g., "Cannot delete comma after nevertheless, as nevertheless does not exist", or "There are multiple occurrences of nevertheless, please specify further"). On the other hand, it previously was hard to see if the speech module correctly interpreted the commanded change because the text was simply replaced. Thus, the interface now temporarily highlights insertions in green, deletions in red (the space at the position), and combinations of green and red for reordering and replacements, where the color fades away 0.5s after the command. That way, the user can quickly see if everything works as expected, or if further corrective commands are required, in which case a simple undo operation can be triggered (as before through the button, hotkey, or by simply saying "*undo*").



(a) Target before speech command.



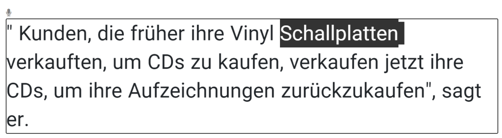
(b) Target after speech command with change highlighting.

Figure 6: Speech command.

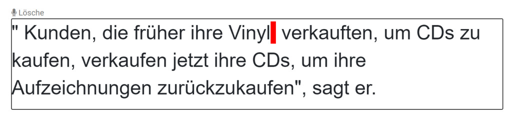
Other ideas we are currently working on include passing the text to the speech recognition to improve transcription results by considering the context (Dymetman et al., 1994) or training the automatic speech recognition towards the user to improve the received transcription.

3.7 Multi-Modal Combination of Pen/Touch/Mouse with Speech

Multi-modal combinations of pen/touch/mouse combined with speech are also supported: Target word(s)/position(s) must first be specified by performing a text selection using the pen, finger touch, or the mouse/keyboard. Afterwards, the user can use a voice command like *"delete"*, *"insert A"*, *"move after/before A/between A and B"*, or *"replace with A"* without needing to specify the position/word, thereby making the commands less complex (see Figure 7).



(a) Target before multi-modal command.



(b) Target after multi-modal command with change highlighting.

Figure 7: Multi-modal command of selection and speech.

In Herbig et al. (2020b), multi-modal interaction received good ratings for insertions and replacements, but worse ratings for reorderings and deletions. One big issue for deletions and reorderings was that multi-word (or partial word) reorder/delete was not supported in the previous implementation; thus, the translator had to place the cursor followed by a speech command multiple times. Due to the possibility of touch selection of multiple (or partial) words, this is now possible using multi-modal combinations of pen/touch/mouse combined with simplified speech commands, thereby hopefully enhancing the user experience. We want to further offer the possibility to keep the selection more straightforward, i.e., allowing the user to place the cursor at one position, but then state, e.g., "delete two words". This should improve situations where speech-only commands are particularly complex due to ambiguities, in which the combined approach was highlighted as advantageous to the speech-only approach. Naturally, the other improvements for the speech case discussed above also work for the multi-modal case, thus hopefully making multi-modal interaction even more natural.

3.8 Eye Tracking

In Herbig et al. (2020b), insertions are the only operation where the multi-modal approach was (non-significantly) faster than speech-only commands, since the position did not have to be verbally specified. We therefore investigate other approaches to enhance the multi-modal case: Apart from improving it by supporting multi-word reorder/delete and simplifying the speech commands as discussed above, we are currently exploring the integration of an eye tracker. The idea is to simply fixate the word to be replaced/deleted/reordered or the gap used for insertion, and state the simplified speech command (e.g., "*replace with A*"/"*delete*"), instead of having to manually place the cursor through touch/pen/mouse/keyboard. Apart from possibly speeding up multi-modal interaction, this approach would also solve the issue reported by several participants in Herbig et al. (2020b) that one would have to "do two things at once", while keeping the advantage of having simple commands in comparison to the speech-only approach.

Our implementation currently only visualizes where the user looks: Upon activation in the Angular client's navigation bar, a request is sent to the node.js server, which launches a Python script for the communication with the eye tracker, and forwards the raw gaze events back to the client. The client can then detect fixations to tell the speech service the current gaze position so that it can be used for multi-modal commands. Figure 8 shows the recognized gaze position.

Apart from combining eye tracking with speech commands, we also plan to combine it with the keyboard, similar to the ReType approach (Sindhwani et al., 2019) but adapted towards the translation and in particular PE domain. Furthermore, we work on memorizing and visu-

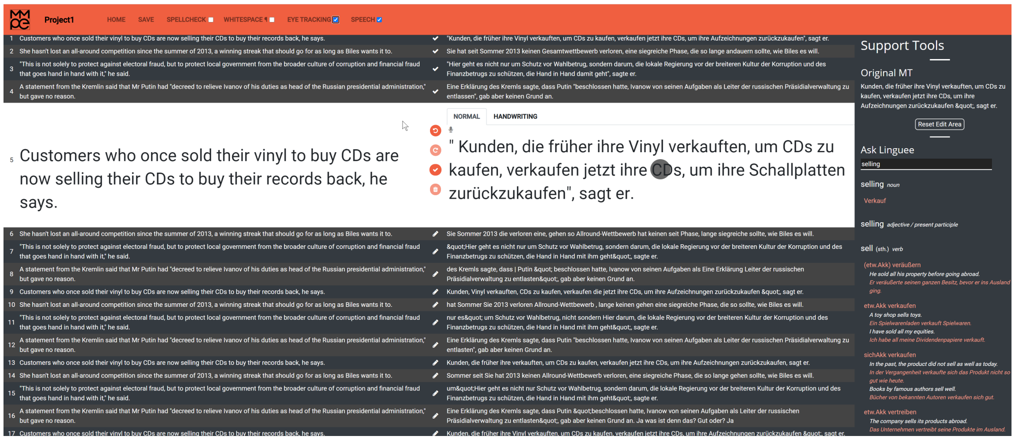


Figure 8: Eye tracking: User gazing at “CDs”.

alizing the last fixation on the source and target views, thereby helping translators navigating through the text not to get lost when switching their attention back and forth between source and target. This approach is similar to GazeMarks (Kern et al., 2010), which has shown its efficiency in visual search tasks with attention shifts.

Our interface facilitates eye tracking, as the large font and screen also enable cheaper models like the Tobii 4C to have sufficient precision to detect words. However, we are currently integrating a more precise eye tracker to explore the differences for the application case.

3.9 Logging

MMPE supports extensive logging functionality: On the one hand, actual keystrokes, touched pixel coordinates, and other events are logged and all UI interactions (like *segmentChange* or *undo/redo/confirm*) are stored, allowing us to analyze the translator’s use of MMPE.

Most importantly, however, we also log all text manipulations on a higher level to simplify text editing analysis. For *insertions*, we log whether a single or multiple words were inserted, and add the actual words and their positions as well as the segment’s content before and after the insertion to the log entry. *Deletions* are logged analogously, and for *reorderings*, we save the old and the new position of the moved word(s) to the log entry. Last, for *replacements*, we log whether only a part of a word was replaced (i.e., changing the word form), whether the whole word was replaced (i.e., correcting the lexical choice), or whether a group of words was replaced. In all cases, the word(s) before and after the change, as well as their positions and the overall segment text, are specified in the log entry (see Figure 9). Furthermore, all log entries contain the modality of the interaction, e.g., speech or pen, thereby allowing the analysis of which modality was used for which editing operation. All log entries with timestamps are created within the client and sent to the server for storage in a JSON file.

We also worked on improvements and extensions to the logging functionality: Apart from bug fixes, we improved logs for copy and paste by adding the clipboard content, better distinguished between delete followed by an insert in comparison to replace operations, improved logs for reordering (distinction into reorder-single, reorder-group, and reorder-partial), and provided more understandable logs for undo/redo. Furthermore, we improved logging for multi-modal commands: We do not merely save whether the interaction was multi-modal, but store whether it was a combination of speech and pen, or speech and mouse, or speech and finger touch. Last, we plan to extend the logging functionality by adding gaze positions, fixations, and

```
{"type":"SPEECH_INPUT_ACTIVATION","interactionSource":"UI","interactionModality":"MOUSE","ts":159611493
0776,"participant":1,"segmentID":5}
{"type":"DELETE_SINGLE","interactionModality":"SPEECH","interactionSource":"MICROPHONE","segmentTextOld
":"\" Kunden, die früher ihre Vinyl verkauften, um CDs zu kaufen, verkaufen jetzt ihre CDs, um ihre
Schallplatten zurückzukaufen\", sagt er.","segmentTextNew":"\" Kunden, die früher ihre verkauften, um
CDs zu kaufen, verkaufen jetzt ihre CDs, um ihre Schallplatten zurückzukaufen\", sagt
er.","position":"char:26","word":"Vinyl","ts":1596114957569,"participant":1,"segmentID":5}
{"type":"INSERT_SINGLE","interactionModality":"PEN","interactionSource":"IINK","segmentTextOld":"\"
Kunden, die früher ihre verkauften, um CDs zu kaufen, verkaufen jetzt ihre CDs, um ihre Schallplatten
zurückzukaufen\", sagt er.","segmentTextNew":"\" Kunden, die früher ihre Schallplatten verkauften, um
CDs zu kaufen, verkaufen jetzt ihre CDs, um ihre Schallplatten zurückzukaufen\", sagt
er.","position":"word6","word":"Schallplatten","ts":1596114997526,"participant":1,"segmentID":5}
{"type":"REPLACE_SINGLE","interactionModality":"KEYBOARD","interactionSource":"MAIN","segmentTextOld":"
\" Kunden, die früher ihre Schallplatten verkauften, um CDs zu kaufen, verkaufen jetzt ihre CDs, um
ihre Schallplatten zurückzukaufen\", sagt er.","segmentTextNew":"\" Kunden, die früher ihre
Schallplatten verkauften, um CDs zu kaufen, verkaufen jetzt ihre CDs, um ihre Vinyl zurückzukaufen\",
sagt er.","position":"word21","wordOld":"Schallplatten","wordNew":"Vinyl","ts":1596115019597,"participa
nt":1,"segmentID":5}
{"type":"SEGMENT_CONFIRM","interactionModality":"MOUSE","interactionSource":"UI","segmentText":"\"
Kunden, die früher ihre Schallplatten verkauften, um CDs zu kaufen, verkaufen jetzt ihre CDs, um
ihre Vinyl zurückzukaufen\", sagt
er.","duration":105170,"ts":1596115022794,"participant":1,"segmentID":5}
{"type":"SEGMENT_SELECT","interactionSource":"UI","interactionModality":"MOUSE","ts":1596115022799,"par
ticipant":1,"segmentID":6}
```


Figure 9: Logging of text manipulations in an easily interpretable granularity.

especially pupil diameter, which can be used for cognitive load analyses (Herbig et al., 2020a). Alongside these, we are planning on integrating further physiological sensors, e.g., the Empatica E4, to capture cognitive load more robustly in a multi-modal fashion by also considering factors like heart rate variability or skin conductance.

4 Conclusion and Future Work

Due to continuously improving MT systems, PE is becoming more and more relevant in modern-day translation. The interfaces used by translators still heavily focus on translation from scratch, and in particular on mouse and keyboard input modalities. Since PE requires less production of text but instead more error corrections, Herbig et al. (2020b) presented and evaluated the MMPE CAT environment that explores the use of speech commands, handwriting input, touch reordering, and multi-modal combinations for PE of MT. In this paper, we use the presented feedback from professional translators to improve and extend the existing prototype: We redesigned the layout, added visualization of whitespaces, fixed issues in hand-writing, allowed multi-word reordering using touch drag and drop and improved its visualization, extended the speech commands, provided better feedback for the user on what the speech commands changed, and improved the logging functionality. Furthermore, we showcased an early implementation of how eye tracking can be integrated, not only for logging but as an actual interaction modality that can be used in combination with speech recognition or the keyboard to quickly correct errors.

As next steps, we want to study how this changed prototype impacts the user experience. Furthermore, we want to finalize the eye-tracking implementation and run a study that specifically explores the combination of eye and speech/keyboard input for PE. Apart from that, longer-term studies exploring how the modality usage changes over time, whether users continuously switch modalities or stick to specific ones for specific tasks, are planned. Furthermore, as eye tracking is already integrated into the prototype for explicit interaction, we want to explore eye-based cognitive load detection and react to high levels of cognitive load by providing alternative MT proposals, as discussed in Herbig et al. (2019b). Finally, to transform MMPE into a fully fledged translation workbench, we want to add user and project management functionality, allow direct loading of common file types like .docx, and cover more language pairs.

References

Alabau, V., Bonk, R., Buck, C., Carl, M., Casacuberta, F., García-Martínez, M., González, J., Koehn, P., Leiva, L., Mesa-Lao, B., et al. (2013). CASMACAT: An open source workbench for advanced computer aided translation. *The Prague Bulletin of Mathematical Linguistics*, 100:101–112.

Alabau, V. and Casacuberta, F. (2012). Study of electronic pen commands for interactive-predictive machine translation. In *Proceedings of the International Workshop on Expertise in Translation and Post-Editing – Research and Application*, pages 17–18.

Aranberri, N., Labaka, G., Diaz de Ilarraza, A., and Sarasola, K. (2014). Comparison of post-editing productivity between professional translators and lay users. In *Proceeding of AMTA Third Workshop on Post-Editing Technology and Practice*, pages 20–33.

Brousseau, J., Drouin, C., Foster, G., Isabelle, P., Kuhn, R., Normandin, Y., and Plamondon, P. (1995). French speech recognition in an automatic dictation system for translators: The TransTalk project. In *Proceedings of Eurospeech Fourth European Conference on Speech Communication and Technology*, pages 193–196.

Carl, M., Jensen, M., and Kristian, K. (2010). Long distance revisions in drafting and post-editing. *CICLing Special Issue on Natural Language Processing and its Applications*, pages 193–204.

Coppers, S., van den Bergh, J., Luyten, K., Coninx, K., van der Lek-Ciudin, I., Vanallemeersch, T., and Vandeghinste, V. (2018). Intellingo: An intelligible translation environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1–13. ACM.

Dymetman, M., Brousseau, J., Foster, G., Isabelle, P., Normandin, Y., and Plamondon, P. (1994). Towards an automatic dictation system for translators: The TransTalk project. In *Proceedings of the ICSLP International Conference on Spoken Language Processing*.

Federico, M., Bertoldi, N., Cettolo, M., Negri, M., Turchi, M., Trombetti, M., Cattelan, A., Farina, A., Lupinetti, D., Martines, A., et al. (2014). The MateCat tool. In *Proceedings of the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 129–132.

Gaspari, F., Toral, A., Naskar, S. K., Groves, D., and Way, A. (2014). Perception vs reality: Measuring machine translation post-editing productivity. In *Third Workshop on Post-Editing Technology and Practice*, page 60.

Green, S., Chuang, J., Heer, J., and Manning, C. D. (2014a). Predictive translation memory: A mixed-initiative system for human language translation. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, pages 177–187. ACM.

Green, S., Heer, J., and Manning, C. D. (2013). The efficacy of human post-editing for language translation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 439–448. ACM.

Green, S., Wang, S. I., Chuang, J., Heer, J., Schuster, S., and Manning, C. D. (2014b). Human effort and machine learnability in computer aided translation. In *Proceedings of the EMNLP Conference on Empirical Methods in Natural Language Processing*, pages 1225–1236.

Herbig, N., Düwel, T., Helali, M., Eckhart, L., Schuck, P., Choudhury, S., and Krüger, A. (2020a). Investigating multi-modal measures for cognitive load detection in e-learning. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pages 88–97.

Herbig, N., Düwel, T., Pal, S., Meladaki, K., Monshizadeh, M., Krüger, A., and van Genabith, J. (2020b). MMPE: A multi-modal interface for post-editing machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1691–1702. Association for Computational Linguistics.

Herbig, N., Pal, S., Düwel, T., Meladaki, K., Monshizadeh, M., Hnatovskiy, V., Krüger, A., and van Genabith, J. (2020c). MMPE: A multi-modal interface using handwriting, touch reordering, and speech commands for post-editing machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 327–334. Association for Computational Linguistics.

Herbig, N., Pal, S., van Genabith, J., and Krüger, A. (2019a). Multi-modal approaches for post-editing machine translation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 231. ACM.

Herbig, N., Pal, S., Vela, M., Krüger, A., and Genabith, J. (2019b). Multi-modal indicators for estimating perceived cognitive load in post-editing of machine translation. *Machine Translation*, 33(1-2):91–115.

Kern, D., Marshall, P., and Schmidt, A. (2010). Gazemarks: Gaze-based visual placeholders to ease attention switching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2093–2102. ACM.

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

Lagoudaki, E. (2009). Translation editing environments. In *MT Summit XII: Workshop on Beyond Translation Memories*.

Läubli, S., Fishel, M., Massey, G., Ehrensberger-Dow, M., and Volk, M. (2013). Assessing post-editing efficiency in a realistic translation environment. In *Proceedings of MT Summit XIV Workshop on Post-Editing Technology and Practice*, pages 83–91.

Martinez, M. G., Singla, K., Tammewar, A., Mesa-Lao, B., Thakur, A., Anusuya, M., Srinivas, B., and Carl, M. (2014). SEECAT: ASR & eye-tracking enabled computer assisted translation. In *The 17th Annual Conference of the European Association for Machine Translation*, pages 81–88. European Association for Machine Translation.

Mesa-Lao, B. (2014). Speech-enabled computer-aided translation: A satisfaction survey with post-editor trainees. In *Proceedings of the EACL 2014 Workshop on Humans and Computer-Assisted Translation*, pages 99–103.

Moorkens, J. (2018). What to expect from neural machine translation: A practical in-class translation evaluation exercise. *The Interpreter and Translator Trainer*, 12(4):375–387.

Moorkens, J. and O'Brien, S. (2015). Post-editing evaluations: Trade-offs between novice and professional participants. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*, pages 75–81.

Moorkens, J. and O'Brien, S. (2017). Assessing user interface needs of post-editors of machine translation. In *Human Issues in Translation Technology*, pages 127–148. Routledge.

O'Brien, S., Moorkens, J., and Vreeke, J. (2014). Kanjingo – a mobile app for post-editing. In *Proceedings of the 17th Annual Conference of the European Association for Machine Translation*.

Schwartz, L., Lacruz, I., and Bystrova, T. (2015). Effects of word alignment visualization on post-editing quality & speed. *Proceedings of MT Summit XV*, 1:186–199.

Sindhwani, S., Lutteroth, C., and Weber, G. (2019). ReType: Quick text editing with keyboard and gaze. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Teixeira, C. S., Moorkens, J., Turner, D., Vreeke, J., and Way, A. (2019). Creating a multimodal translation tool and testing machine translation integration using touch and voice. *Informatics*, 6.

Theologitis, D. (1998). Language tools at the EC translation service: The theory and the practice. In *Proceedings of the 20th Conference Translating and the Computer*, pages 12–13.

Toral, A., Wieling, M., and Way, A. (2018). Post-editing effort of a novel with statistical and neural machine translation. *Frontiers in Digital Humanities*, 5:9.

Torres-Hostench, O., Moorkens, J., O'Brien, S., Vreeke, J., et al. (2017). Testing interaction with a mobile MT post-editing app. *Translation & Interpreting*, 9(2):138.

van den Bergh, J., Geurts, E., Degraen, D., Haesen, M., van der Lek-Ciudin, I., Coninx, K., et al. (2015). Recommendations for translation environments to improve translators' workflows. In *Proceedings of the 37th Conference Translating and the Computer*, pages 106–119. Tradulex.

Vandeghinste, V., Vanallemeersch, T., Augustinus, L., Bulté, B., Van Eynde, F., Pelemans, J., Verwimp, L., Wambacq, P., Heyman, G., Moens, M.-F., et al. (2019). Improving the translation environment for professional translators. *Informatics*, 6(2):24.

Vandeghinste, V., Vanallemeersch, T., Augustinus, L., Pelemans, J., Heyman, G., van der Lek-Ciudin, I., Tezcan, A., Degraen, D., van den Bergh, J., Macken, L., et al. (2016). Scate – Smart Computer-Aided Translation Environment. *Baltic Journal of Modern Computing*, 4(2):382–382.

Vela, M., Pal, S., Zampieri, M., Naskar, S. K., and van Genabith, J. (2019). Improving CAT tools in the translation workflow: New approaches and evaluation. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 8–15.

Wallis, J. (2006). *Interactive Translation vs Pre-translation in the Context of Translation Memory Systems: Investigating the Effects of Translation Method on Productivity, Quality and Translator Satisfaction*. PhD thesis, University of Ottawa.

Yamada, M. (2015). Can college students be post-editors? An investigation into employing language learners in machine translation plus post-editing settings. *Machine Translation*, 29(1):49–67.

Zampieri, M. and Vela, M. (2014). Quantifying the influence of MT output in the translators' performance: A case study in technical translation. In *Proceedings of the EACL 2014 Workshop on Humans and Computer-Assisted Translation*, pages 93–98.

Zapata, J. (2016). Translating on the go? Investigating the potential of multimodal mobile devices for interactive translation dictation. *Tradumàtica: Traducció i Tecnologies de la Informació i la Comunicació*, 1(14):66–74.

Zapata, J., Castilho, S., and Moorkens, J. (2017). Translation dictation vs. post-editing with cloud-based voice recognition: A pilot experiment. *Proceedings of MT Summit XVI*, 2.

Zaretskaya, A. and Seghiri, M. (2018). *User Perspective on Translation Tools: Findings of a User Survey*. PhD thesis, University of Malaga.

Zaretskaya, A., Vela, M., Pastor, G. C., and Seghiri, M. (2016). Comparing post-editing difficulty of different machine translation errors in Spanish and German translations from English. *International Journal of Language and Linguistics*, 3(3):91–100.