A Process Study of Computed Aided Translation

Philipp Koehn (pkoehn@inf.ed.ac.uk)
School of Informatics, University of Edinburgh

September 28, 2009

Abstract. We investigate novel types of assistance for human translators, based on statistical machine translation methods. We developed the computer aided tool *Caitra* that makes suggestions for sentence completion, shows word and phrase translation options, and allows postediting of machine translation output. We carried out a study of the translation process that involved non-professional translators that were native in either French or English and recorded their interaction with the tool. Users translated 192 sentences from French news stories into English. Most translators were faster and better when using assistance. A detailed examination of the logs also provides insight into the human translation process, such as time spent on different activities and length of pauses.

Keywords: computer aided translation, interactive translation, translation process study, statistical machine translation

1. Introduction

Today's machine translation systems are mostly used for inbound translation (also called assimilation), where the reader accepts lower quality translation for instant access to foreign language text. Most prominent is Google Translate[1] which is freely available on the web. However, the demands for quality are much higher for outbound translation (also called dissemination), where the reader is typically an unsuspecting customer or citizen who is seeking information about products or services. Since machine translation alone cannot meet these demands, human translators are required for such high-quality publication-ready translation. This creates opportunities for computer aided translation tools that aim to improve the productivity of human translators.

While machine translation has made tremendous progress in recent years, this progress has made few inroads into tools for human translators. Although it has become frequent practice in the industry to provide human translators with machine translation output for postediting, typically no deeper integration of machine translation and human translation is found in translation agencies.

An interesting approach was pioneered by the TransType project (Langlais et al., 2000a). Here, the machine translation system makes

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

sentence completion predictions in an interactive machine translation setting. The users may accept them or override them by typing in their own translations, which triggers new suggestions by the tool (Barrachina et al., 2009). But other information of the machine translation system may also be useful for the human translator, such as alternative translations for the input words and phrases.

We developed the web-based translation tool Caitra (Koehn, 2009) that implements various types of assistance. We report on a study on involving ten human translators, whose interaction with the tool was logged in great detail. The task was the translation of news stories from French–English which is a relatively easy task since the users are familiar with the general content and French–English machine translation quality is quite high (Koehn and Haddow, 2009; Callison-Burch et al., 2009). Our study showed that most translators were able to produce translations both faster and better with such assistance.

The detailed log also allowed us to explore what translators spend their time on, and how this changes when assistance is given. We were especially interested how much time translators spend on the activities like typing or mouse clicks and how much on pauses of different lengths.

2. Related Work

Current tools for translators focus on the use of translation memories that retrieve matching input sentence or similar input sentences (fuzzy matches) and present them to the user. Such tools are widespread and offered by commercial vendors such as SDL Trados [2] or by open source projects, for instance OmegaT[3]. The wikiBABEL project developed a tool similar to ours, which starts with machine translation output for post-editing and offers additional help in form of translation dictionaries (Kumaran et al., 2008). In the meantime, Google has started offering a similar basic service[4] (Galvez and Bhansali, 2009).

In 1990s, the increase computer use by human translators has enabled process studies of translation (Fraser, 1996) based on user activity data. A widely used tool in the research community is Translog which logs keystrokes of translators (Jakobsen and Schou, 1999). This allows the collection of timing statistics, but also data about revision ratios, i.e. the relative amount of deletions and cursor movements to final characters, for further analysis (Buchweitz and Alves, 2006).

[2] http://www.trados.com/

[3] http://www.omegat.org/

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

This methodology to analyze the translation process was also applied to interactive translation within the TransType project (Langlais et al., 2000b) and for postediting machine translation (O'Brien, 2005).

Studies may also make use of *think aloud protocols* (Jääskeläinen, 2001), often referred to as TAP, in which the translator narrates the thought process behind her actions. However, as for instance Jakobsen, 2003 (Jakobsen, 2003) points out, this narration has an significant effect on the translation process, especially translation speed and the amount of segmentation.

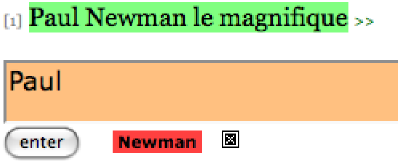
The combination of TAP and key logs allows the detection of translation problems (indicated by pauses) and their identification. Jensen and Jakobsen, 2000 (Jensen and Jakobsen, 2000) use six types of problem solving activities and note that on average, translators solve two of them per minute. They distinguish between dictionary lookup, direct (literal) methods such as borrowing, loan translation, and literal translation on the one hand and indirect methods such as paraphrasing and adaptation on the other. Similarly, Lörscher, 2005 (Lörscher, 2005) identifies 22 elements of translation strategies that include actions (such as reading the source text or paraphrasing the source text) which are combined in larger structures of activity.

A third source of information about the translation process comes from eye tracking. Sharmin et al., 2008 (Sharmin et al., 2008) examine the amount of time translators look at the source text and target text and found that slightly more time is spent in looking at the target text. They also found that touch typist are not faster translators than ones that have to look at the keyboard and examined how text difficulty and time pressure affects fixations. Eye tracking also allows to pin-point which part in the source text the translator looks at when pausing for an extended time and hence the identification of difficult parts within it.(Carl et al., 2008).

In our process study we were mainly concerned with translation speed. Also, when different types of assistance are offered, we were interested how these were utilized. Compared to the cited studies, we use a larger corpus and a larger number of test subjects. This is also the first direct comparison of post-editing and interactive machine translation methods.

3. Types of Assistance

Our translation tool Caitra is implemented in Ruby on Rails (Raymond, 2007) as a web-based client-server architecture using Ajax Web 2.0 technologies. The machine translation back-end is powered by the



[1] Paul Newman le magnifique >>

Paul

Figure 1. **Interactive Machine Translation.** Caitra uses the search graph of the machine translation decoder to suggest words and phrases to continue the translation.

Moses decoder (Koehn et al., 2007). The tool is delivered over the web to allow for easier user studies with remote users, but also to expose the tool to a wider community to gather additional feedback. You can find Caitra online at `http://www.caitra.org/`.

The tool allows the uploading of documents using a simple text box. This text is then processed by a back-end job to pre-compute all the necessary data (machine translation output, translation options, search graphs). This process takes a few minutes.

Finally, the user is presented with an interface that includes all the different types of assistance. Each may be turned off, if the user finds it distracting. The user translates one sentence at a time, while its context (both input and user translation, including the preceding and following paragraph) is displayed for reference.

In the next three sections, we will describe each type of assistance in detail.

3.1. Prediction

In the sentence-completion paradigm, the human translator is still in charge of creating the translation word by word, but she is aided by a machine translation system that interactively makes suggestions for completing the sentence, and updates these suggestions based on her input. The scenario is very similar to the auto-completion function for words, search terms, email addresses, etc. in modern office applications or predictive text entry in mobile phones.

See Figure 1 for a screenshot of the incarnation of this method in Caitra. The user is given an input sentence and a standard web text box to type in her translation. In addition, the system makes suggestions about the next word (or phrase) to be added to the translation. The user may accept this (by pressing the TAB key), or type in her own translation. The tool updates the prediction based on the user input.

The predictions are based on a statistical machine translation system. Given the input and the partial translation of the user, the machine translation system computes the optimal translation of the input sentence, constrained by matching the user input (Och et al., 2003). The predicted translation is shown to the user in form of short phrases (mirroring the underlying phrase-based statistical translation model).

In contrast to traditional work on interactive machine translation, the displayed suggestion consists of only very few words to not overload the reading capacity of the user. We have not yet carried out studies to explore the optimal length of suggestions, or even when not to provide suggestions at all, in cases when they will be most likely useless and distracting. See work by Foster et al., 2002 (Foster et al., 2002) on prediction length in the TransType project.

We store the search graph produced by the machine translation decoder in a database. During the user interaction, we quickly match user input against the graph using a string edit distance measure. The prediction is the optimal completion path that matches the user input with (a) minimal string edit distance and (b) highest sentence translation probability. This computation takes place at the server and is implemented in C++. While Caitra only displays one phrase prediction at a time, the entire completion path is transmitted to the client. Acceptance of a system suggestion will instantly lead to another suggestion, while typed-in user translations require the computation of a new sentence completion path. This typically takes less than a second.

See Figure 2 for the pseudo-code of the algorithm that matches user input (also called the prefix) against the search graph. Matching is very fast, if the prefix is found verbatim in graph, or if only very few edits are needed. Hence, the algorithm first tries to match the prefix allowing no errors. While that fails, the number of allowable errors is increased iteratively by 1.

The algorithm associates with each state of the graph backpointers that point back to the cheapest error and cost path with which it can be reached. There are multiple backpointers for each state, since the state may match the prefix at different positions. When examining each state's backpointers, all forward transitions are examined using a string edit distance between the remaining prefix and the words in the transition phrase (line 9). This may consume the remaining prefix, and possibly lead to a new best path for the corresponding error-level (line 10–14). Otherwise, new backpointers for the forward states are created (line 15–25).

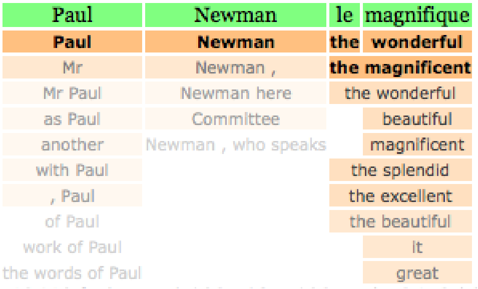
The required information for this algorithm is derived straightforwardly from the search graph of the statistical machine translation decoder. In practice it runs very fast, most often in a small fraction of a second.

```
Input: user prefix $u$, search graph $g$
Output: best path $p$
allowable error $e = 0$
best path $p_i = \{\}$ for all error $i$
add backpointer ( cost=0.0, error=0, toProcess=$u$ ) to start state
while best path $p_{e-1} == \{\}$ and error $e <$ length($p$) do
    for all state $s \in g$ in topologically increasing order do
        for all backpointer $b$ of state $s$ do
            if $b$.error $== e$ then
                for all transition $t$ from state $s$ do
                    compute string edit distance matrix for $b$.toProcess, $t$.phrase
                    for all matches $m$ in matrix that consumed all of $b$.toProcess do
                        new cost $c_n = s$.cost + $t$.cost + $t$.toState.forwardCost
                        new error $e_n = s$.error + $m$.error
                        if $c_n < p_{e_n}$.cost then set this as $p_{e_n}$
                    end for
                    for all matches $m$ in the matrix that consumed all of $t$.phrase do
                        reached new state $s_n = t$.toState
                        create new backpointer $b_n$
                        $b_n$.cost=$s$.cost+$t$.cost
                        $b_n$.error=$s$.error+$t$.error
                        $b_n$.toProcess=$s$.toProcess-$t$.phrase
                        $b_c$ = current backpointer for state $s_n$ at prefix pos. $b_n$.toProcess
                        if $b_c$ not defined or $b_n$.error $<$ $b_c$.error or $b_n$.error $==$ $b_c$.error
                        and $b_n$.cost $<$ $b_c$.cost then
                            make $b_n$ new backpointer for state $s_n$, pos. $b_n$.toProcess
                        end if
                    end for
                end for
            end if
        end for
    end for
end while
best path $p = p_e$
```

Figure 2. **Finding the best match for a prefix in a search graph.** The worst case complexity of the algorithm is linear in the number of states and quadratic in the length of the prefix (given finite limits on state fan-out and phrase lengths), in practice it is much faster.

3.2. Options from the Translation Table

Phrase-based statistical machine translation methods store their translation knowledge in form of a phrase translation table that was automatically acquired from large amounts of translated text. For each input word or input word sequence, this translation table is consulted for the most likely translation options. A heuristic beam search algorithm explores these options and their ordering to find the most



Paul	Newman	le	magnifique
Paul	**Newman**	**the**	**wonderful**
Mr	Newman ,	**the magnificent**	
Mr Paul	Newman here	the wonderful	
as Paul	Committee		beautiful
another	Newman , who speaks		magnificent
with Paul		the splendid	
, Paul		the excellent	
of Paul		the beautiful	
work of Paul			it
the words of Paul			great

Figure 3. **Translation Options.** The most likely word and phrase translation are displayed alongside the input words, ranked and color-coded by their probability.

likely sentence translation (which takes into account various scoring functions, such as the use of an n-gram language model).

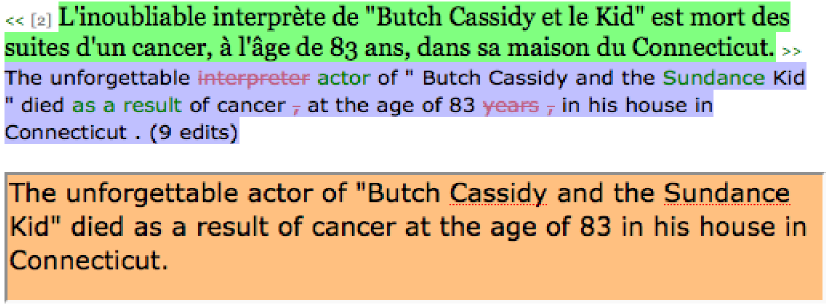
These translation options may also be of interest to a human translator, so we display them in Caitra. See Figure 3 for an example. For instance, the tool suggests for the translation of the French *magnifique* the English options *wonderful, beautiful, magnificent,* and *great*, among others. The user may click on any of these phrases and it is added into the text box. The user may also just glance at these suggestions and then type in the translation herself. The options are color-coded and ranked based on their score. Note that since these options are extracted from a translated corpus using various automatic methods, often inappropriate translations are included, such as the translation of *Newman* into *Committee*.

For each translation option a score is computed to assess its utility. This score is the sum of

- future cost estimate of the phrase
- outside cost estimate for the remaining sentence

This number allows the ranking of words vs. phrases of different length. The ranking of the phrases never places a lower scoring option above a higher scoring option. The absolute score is used to color code the options. Up to ten table rows are filled with options.

Since the user may click on the options, or may simply type in translations inspired by the options, it is not straightforward to evaluate their usefulness. Experience so far suggests that the options help novice users with unknown words and advanced users with suggestions that are not part of their active vocabulary. It may be possible that these options even allow users that do not know the source language to translate, as in work done by Albrecht et al., 2009 (Albrecht et al., 2009).



<< [2] L'inoubliable interprète de "Butch Cassidy et le Kid" est mort des suites d'un cancer, à l'âge de 83 ans, dans sa maison du Connecticut. >>

The unforgettable ~~interpreter~~ actor of " Butch Cassidy and the Sundance Kid " died as a result of cancer ~~,~~ at the age of 83 ~~years~~ ~~,~~ in his house in Connecticut . (9 edits)

The unforgettable actor of "Butch Cassidy and the Sundance Kid" died as a result of cancer at the age of 83 in his house in Connecticut.

Figure 4. **Postediting Machine Translation.** Starting with the sentence translation of the machine translation system, the user edits it and the tool indicates changes.

3.3. Postediting Machine Translation

The provision of a full sentence translation from the machine translation system is trivial compared to the other types of assistance. When a user starts a new sentence using this aid, the text box already contains the machine translation output and the user only makes changes to correct errors.

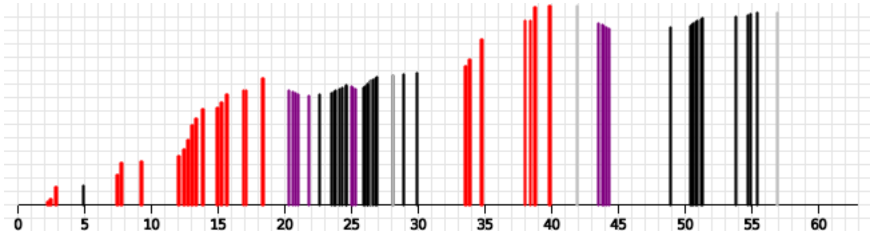
See Figure 4 for an example. Caitra also compares the user's translation in form of string edit distance against the machine translation. This is illustrated above the text box, to possibly alert the user to mistakenly dropped or added content.

4. User Study

Caitra tracks every key stroke and mouse click of the user, which then allows for a detailed analysis of the user's interaction with the tool. See Figure 5 for a graphical representation of the user activity during the translation of a sentence. The graph plots sentence length (in characters) against the progression of time.

4.1. Experimental Design

We recruited 10 human translators for our study. Half of the translators are native speakers of French (L2) studying at the University of Edinburgh in Scotland, the other half native speakers of English (L1) with university-level French skills. None of the participants were professional translators, either practicing or in training to be. In the following, the translators are referred to as L2a, L1a, L2b, L1b, and



Input: "Un échange de coups de feu s'est produit, et la moitié des ravisseurs ont été tués, les autres s'enfuyant", a dit ce responsable qui a requis l'anonymat.

MT: "A exchange of fire occurred, and half of the kidnappers were killed, the other is enfuyant," said this official who has requested anonymity.

User: "An exchange of fire occurred, and half of the kidnappers were killed, the others running away", said the source who has requested anonymity.

Figure 5. **User Activity.** The graph plots the time spent on translation (in seconds, x-axis) against the length of the sentence (y-axis) with color-coded activities (bars). Bars indicate the sentence length at each point in time when a user action takes place. Acceptance of predictions are red, DEL key strokes purple, key strokes for cursor movement grey, and key strokes that add characters are black. The user first slowly accepted the interactive machine translation predictions (second 0-12), then more rapidly (second 12-20), followed by a period of deletions and typing that did not make the translation longer (second 20-30). After a short pause, predictions were accepted again (second 33-40), followed by deletions and typing (second 40-57).

so on. All translators are associated with the University of Edinburgh, being either students or staff. The delivery of the translation tool over the web allowed the translators to work at their own convenience within a two week period. They were rewarded for their efforts with a fixed amount of money instead of an hourly wage to give them an incentive to be productive.

Each translator translated the same set of documents with the total size of 192 sentences from French to English. The document set was taken from the 2009 EACL Workshop on Statistical Machine Translation and consists of news paper articles from *Le Devoir*, *Le Figaro*, *Les Echos* and *Libération*.

The text is broken up into five blocks of about 40 sentences and 1000 words. Each block consists of one to three complete documents each. Table I gives details about the blocks and their distribution to the translators under the five different types of assistance: (1) unassisted, (2) postediting machine translation output, (3) options from the translation table, (4) prediction (sentence completion), and (5) options and predictions.

While it is not possible to give the same block to the same translator with different types of assistance, we distributed them in a way that

Table I. **Permutation of Assignments.**
Translation blocks A–E are assigned to the human translators a–e to translate under varying types of assistance. Averaging over all translators, translation time differs slightly from 3.1–3.9 seconds/word.

Block	Time	Doc's	Sentences	Words	Sources
A	3.9 sec/word	2	32	925	Le Devoir, Les Echos
B	3.4 sec/word	2	35	929	Le Devoir, Libération
C	3.7 sec/word	3	39	1105	Les Echos (2), Libération
D	3.1 sec/word	1	46	1418	Le Devoir
E	3.2 sec/word	2	40	1108	Libération (2)

Block	a	b	c	d	e
A	Unassisted	Opt.+Pred.	Prediction	Options	Postedit
B	Postedit	Unassisted	Opt.+Pred.	Prediction	Options
C	Options	Postedit	Unassisted	Opt.+Pred.	Prediction
D	Prediction	Options	Postedit	Unassisted	Opt.+Pred.
E	Opt.+Pred.	Prediction	Options	Postedit	Unassisted

each block is translated by each type of translator (L2/L1) under each condition. One concern is that different blocks pose different degrees of difficulty. This is true to some extent in our data set, where the average translation time for the five blocks varies from 3.1 to 3.9 seconds per word. However, it is not clear if the slow translation of a block is due to the difficulty of the block or if a individual translator is particularly ill-equipped to translate it.

4.2. Evaluation

Since Caitra logs the time spent on each sentence, it is straightforward to compute the average time per input word which we use as our evaluation of translation speed.

Speed is not the only criterion of success, the translations have to be correct as well. Evaluation of translation quality is a difficult problem, since ten different translators will almost always produce ten different translations, and it hard to assess which ones are correct.

We relied on human judges to check each translation. Given the French source sentence in context (two preceeding and two following sentences), they were asked to classify translations as correct with the following instructions:

> *Indicate whether each user's input represents a fully fluent and meaning-equivalent translation of the source. The source is shown with context, the actual sentence is bold.*

A web-based evaluation tool was deployed to solicit these judgements. All ten translations for each sentence were displayed on the same screen. The judges were fluent in both French and English. Sentences were randomly distributed to judges, so the number of judgments per sentence varies. On average, each sentence (and each of its translations) was evaluated about five times.

5. Results and Analysis

The detailed logs of the translator actions offer a wealth of data. We are not only interested in translation speed and quality, but we would also like to gain some insight into the translation process and the behavior of the translators.

5.1. Speed and Quality

The most important questions from the view of the tool developer are: do human translators produce better translations and are they faster than when unassisted? The short answer is: mostly, yes.

Table II gives a slightly longer answer. On average, the human translators are faster and also achieve better translation quality using any type of assistance offered. Only in very few instances, they are both slower and worse. Individual results vary, see the table for details. Translators are fastest with postediting and obtain highest translation performance when postediting and using prediction and options.

When postediting, 8 translators are faster and better, when using the options 4 translators are faster and better, when using the predictions 6 translators are faster and better, and when using both predictions and options 6 translators are faster and better. 4 Translators are faster and better with all of the assistances offered, and only two translators achieved no gains in both dimensions with any assistance.

A note on the quality judgments: We were surprised by the low correctness numbers we obtained from the human judges (the overall average is 50%). When using this metric in machine translation evaluation, human reference translations were judged 85-90% correct using the same metric. After querying some of the human judges, we were left with the impression that they were overly critical (*"this translation sounds funny to me"*), and may also be tempted, when given 10 translations at a time, to label half of them as correct and the other half as wrong — an implicit ranking of the translations.

Table II. **Speed and Quality.**
On average, translators are faster and also achieve better translation quality using any of the assistances offered. Individual results vary.

User	**Unassisted**	**Postedit**	**Options**	**Prediction**	**Pred.+Opt.**
L2a	3.3sec/word 23% correct	**1.2s (-2.2s)** **39% (+16%)**	**2.3s (-1.0s)** **45% (+22%)**	**1.1s (-2.2s)** **30% (+7%)**	**2.4s (-0.9s)** **44% (+21%)**
L2b	7.7sec/word 35% correct	**4.5s (-3.2s)** **48% (+13%)**	**4.5s (-3.3s)** **55% (+20%)**	**2.7s (-5.1s)** **61% (+26%)**	**4.8s (-3.0s)** **41% (+6%)**
L2c	3.9sec/word 50% correct	**1.9s (-2.0s)** **61% (+11%)**	**3.8s (-0.1s)** **54% (+4%)**	**3.1s (-0.8s)** **64% (+14%)**	**2.5s (-1.4s)** **61% (+11%)**
L2d	2.8sec/word 38% correct	**2.0s (-0.7s)** **46% (+8%)**	2.9s (+0.1s) 59% (+21%)	2.4s (-0.4s) 37% (-1%)	**1.8s (-1.0s)** **45% (+7%)**
L2e	5.2sec/word 58% correct	**3.9s (-1.3s)** **64% (+6%)**	4.9s (-0.2s) 56% (-2%)	**3.5s (-1.7s)** **62% (+4%)**	4.6s (-0.5s) 56% (-2%)
L1a	5.7sec/word 16% correct	**1.8s (-3.9s)** **50% (+34%)**	**2.5s (-3.2s)** **34% (+18%)**	**2.7s (-3.0s)** **40% (+24%)**	**2.8s (-2.9s)** **50% (+34%)**
L1b	3.2sec/word 64% correct	2.8s (-0.4s) 56% (-8%)	*3.5s (+0.3s)* *60% (-4%)*	*6.0s (+2.8s)* *61% (-3%)*	*4.6s (+1.4s)* *57% (-7%)*
L1c	5.8sec/word 52% correct	**2.9s (-3.0s)** **53% (+1%)**	4.6s (-1.2s) 37% (-15%)	**4.1s (-1.7s)** **59% (+7%)**	**2.7s (-3.1s)** **53% (+1%)**
L1d	3.4sec/word 49% correct	3.1s (-0.3s) 49% (+0%)	4.3s (+0.9s) 51% (+2%)	3.8s (+0.4s) 53% (+4%)	3.7s (+0.3s) 58% (+9%)
L1e	2.8sec/word 68% correct	**2.6s (-0.2s)** **79% (+11%)**	*3.5s (+0.7s)* *59% (-9%)*	2.8s (-0.0s) 64% (-4%)	*3.0s (+0.2s)* *66% (-2%)*
avg.	4.4sec/word 47% correct	**2.7s (-1.7s)** **55% (+8%)**	**3.7s (-0.7s)** **51% (+4%)**	**3.2s (-1.2s)** **54% (+7%)**	**3.3s (-1.1s)** **53% (+6%)**

See Figure 6 for two sentences, their translations, and the quality judgments of each translation. For some sentences the judges disagree — for instance 4/2 indicates that four judges deemed a translation to be correct while two labeled it as wrong. Also note that each translator came up with a different translation, a common observation in human translation.

Src.	C'est un groupe d'élus républicains qui avait fait capoter le premier projet d'entente, la semaine dernière.
MT	It is a group of elected Republicans that wrecked the first draft agreement last week.
5/1	It is a group of elected Republicans who failed the first draft of the understanding last week. *(Options, L1a)*
5/1	It is a group of elected Republicans that wrecked the first draft agreement, last week. *(Prediction, L1b)*
4/2	It is a group of elected Republicans that wrecked the first draft of understanding last week. *(Prediction+Options, L1c)*
0/6	The first to propose a rescue package last week was a group of Republican representatives. *(Unassisted, L1d)*
5/1	A group of elected Republicans wrecked the agreement's first draft last week. *(Postedit, L1e)*
6/0	It is a group of elected Republicans that wrecked the first draft of the agreement last week. *(Options, L2a)*
6/0	It's a group of elected Republicans that wrecked the first draft agreement last week. *(Prediction, L2b)*
4/2	It is a group of Republican representatives which had wrecked the first draft of an understanding last week. *(Prediction+Options, L2c)*
3/3	It was a group of elected republicans which had made the first agreement project fail last week. *(Unassisted, L2d)*
5/1	A group of elected Republicans has already wrecked the first draft agreement last week. *(Postedit, L2e)*

Src.	Sans se démonter, il s'est montré concis et précis.
MT	Without dismantle, it has been concise and accurate.
1/3	Without fail, he has been concise and accurate. *(Prediction+Options, L1a)*
4/0	Without getting flustered, he showed himself to be concise and precise. *(Unassisted, L1b)*
4/0	Without falling apart, he has shown himself to be concise and accurate. *(Postedit, L1c)*
1/3	Unswayable, he has shown himself to be concise and to the point. *(Options, L1d)*
0/4	Without showing off, he showed himself to be concise and precise. *(Prediction, L1e)*
1/3	Without dismantling himself, he presented himself consistent and precise. *(Prediction+Options, L2a)*
2/2	He showed himself concise and precise. *(Unassisted, L2b)*
3/1	Nothing daunted, he has been concise and accurate. *(Postedit, L2c)*
3/1	Without losing face, he remained focused and specific. *(Options, L2d)*
3/1	Without becoming flustered, he showed himself concise and precise. *(Prediction, L2e)*

Figure 6. **Examples of translations and their evaluation.** Several judges labeled translations as correct/wrong, the figure lists the number of such judgments for each sentence.

5.2. Utilizing Assistance

Let us now take a closer look at how translators used the assistance offered to them.

The log of each sentence translation is a sequence of events (key strokes, clicks) at specific time points. We would like to characterize broader activities, such as *typing* or *pauses*, and break up the very detailed sequence of actions into larger intervals of such activities.

We define an activity as a time interval, in which we observe specific events. For instance, the activity of typing is an interval of time that only consists of keystrokes without any significant pauses and no other event. By *significant pause*, we imply that the window of one second before a keystroke and one second after a keystroke is part of the typing activity, and only periods lacking such activities are labeled as pauses.

Definition: Activity. Each event e has a timepoint $t(e)$ and a type $y(e) \in Y = \{\text{key, click, tab}\}$. Let L be the set of all events for the translation of a sentence, and w the window size (one second). We define an activity is an interval $I = [t_1, t_2]$ of the type $A \subset Y$ as

$$\begin{aligned} & I[t_1,t_2] \text{ has type } A \Leftrightarrow \\ & \qquad \forall e \in L : t_1 - w \leq t(e) \leq t_2 + w \rightarrow y(e) \in A \\ & \quad \text{and } \ \forall y \in A, t \in I : \exists e \in L : y(e) = y, t - w \leq t(e) \leq t + w \end{aligned} \tag{1}$$

Under this definition, the period of translating a sentence segments into a unique sequence of maximal intervals of activities (meaning, no neighboring intervals have the same activity).

The set of different activity types is a power set of the types of events, but we collapse all activities with multiple types of events into one type: the *mixed* activity. We further break up pauses into

- 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

Note that there are no pauses shorter than 2 seconds, since these are necessarily part of non-pause activities.

We are less interested in the number of intervals, but rather how much time is spent on each type of activity. Does the translator spend most of her time in big pauses, or on typing keys? Table III gives a breakdown for each translator for each type of assistance. The timing

Table III. **Time Spent on Activities.**
We break down user actions into a sequence of intervals of specific activities: pause (initial, end, short, medium, big), key strokes, clicking on options, TAB key strokes to accept predictions, and mixed activities (key/tab/click within the same interval). The table shows how much time (measured as seconds per input word) is spent on each activity.

User: Q2a	total	initp	endp	shortp	midp	bigp	key	click	tab	mixed
Unassisted	3.31s	0.07s	0.11s	0.18s	1.04s	0.07s	1.84s	-	-	-
Postedit	1.16s	0.48s	0.08s	0.05s	0.27s	-	0.27s	-	-	-
Options	2.28s	0.19s	0.09s	0.32s	0.62s	-	0.34s	0.68s	-	0.04s
Prediction	1.11s	0.04s	0.02s	0.07s	0.22s	-	0.27s	-	0.42s	0.06s
Pred.+Opt.	2.38s	0.13s	0.12s	0.22s	0.73s	-	0.60s	0.27s	0.25s	0.07s

User: Q2b	total	initp	endp	shortp	midp	bigp	key	click	tab	mixed
Unassisted	7.74s	1.29s	0.11s	0.25s	1.83s	1.94s	2.32s	-	-	-
Postedit	4.50s	1.47s	0.43s	0.14s	0.95s	0.41s	1.09s	-	-	-
Options	4.46s	0.59s	0.11s	0.36s	0.85s	0.70s	1.46s	0.38s	-	0.01s
Prediction	2.67s	0.29s	0.27s	0.19s	0.74s	0.09s	0.63s	-	0.41s	0.05s
Pred.+Opt.	4.79s	0.58s	0.35s	0.41s	1.31s	0.48s	0.89s	0.47s	0.24s	0.04s

User: Q2c	total	initp	endp	shortp	midp	bigp	key	click	tab	mixed
Unassisted	3.88s	0.23s	0.16s	0.33s	0.71s	-	2.45s	-	-	-
Postedit	1.92s	0.59s	0.16s	0.10s	0.49s	-	0.57s	-	-	-
Options	3.77s	0.36s	0.19s	0.55s	0.88s	-	1.15s	0.58s	-	0.07s
Prediction	3.11s	0.20s	0.27s	0.38s	0.46s	-	1.28s	-	0.44s	0.07s
Pred.+Opt.	2.53s	0.27s	0.18s	0.41s	0.29s	-	0.71s	0.56s	0.02s	0.08s

User: Q2d	total	initp	endp	shortp	midp	bigp	key	click	tab	mixed
Unassisted	2.79s	0.23s	0.04s	0.20s	0.39s	0.14s	1.78s	-	-	-
Postedit	2.05s	0.53s	0.15s	0.10s	0.50s	0.23s	0.56s	-	-	-
Options	2.89s	0.13s	0.12s	0.30s	0.50s	-	1.83s	0.01s	-	0.00s
Prediction	2.38s	0.18s	0.11s	0.29s	0.36s	0.08s	0.73s	-	0.60s	0.03s
Pred.+Opt.	1.78s	0.13s	0.11s	0.23s	0.18s	-	0.50s	0.00s	0.60s	0.04s

User: Q2e	total	initp	endp	shortp	midp	bigp	key	click	tab	mixed
Unassisted	5.17s	0.28s	0.04s	0.33s	1.86s	0.48s	2.18s	-	-	-
Postedit	3.87s	0.76s	0.08s	0.22s	0.94s	0.73s	1.15s	-	-	-
Options	4.94s	0.28s	0.10s	0.56s	1.36s	0.38s	1.99s	0.26s	-	0.02s
Prediction	3.46s	0.19s	0.04s	0.40s	0.89s	0.14s	1.19s	-	0.53s	0.08s
Pred.+Opt.	4.64s	0.18s	0.10s	0.55s	1.02s	0.46s	2.03s	0.06s	0.23s	0.02s

Table III.
(continued from previous page)

User: Q1a	total	initp	endp	shortp	midp	bigp	key	click	tab	mixed
Unassisted	5.68s	0.54s	0.12s	0.31s	1.78s	0.71s	2.21s	-	-	-
Postedit	1.82s	0.66s	0.10s	0.09s	0.46s	0.20s	0.31s	-	-	-
Options	2.46s	0.36s	0.13s	0.25s	0.60s	0.12s	0.24s	0.73s	-	0.03s
Prediction	2.70s	0.32s	0.20s	0.14s	0.80s	0.43s	0.48s	-	0.26s	0.06s
Pred.+Opt.	2.82s	0.21s	0.42s	0.17s	1.20s	0.07s	0.44s	0.13s	0.16s	0.02s
User: Q1b	total	initp	endp	shortp	midp	bigp	key	click	tab	mixed
Unassisted	3.19s	0.14s	0.07s	0.23s	0.43s	0.08s	2.24s	-	-	-
Postedit	2.84s	0.76s	0.20s	0.16s	0.81s	0.13s	0.78s	-	-	-
Options	3.50s	0.21s	0.13s	0.39s	1.03s	0.07s	0.98s	0.62s	-	0.07s
Prediction	5.97s	0.60s	0.21s	0.55s	1.30s	0.49s	2.82s	-	-	-
Pred.+Opt.	4.64s	0.38s	0.31s	0.61s	1.74s	0.07s	0.46s	1.00s	-	0.07s
User: Q1c	total	initp	endp	shortp	midp	bigp	key	click	tab	mixed
Unassisted	5.82s	0.27s	0.15s	0.52s	1.51s	0.26s	3.11s	-	-	-
Postedit	2.86s	0.61s	0.32s	0.16s	1.02s	0.11s	0.64s	-	-	-
Options	4.60s	0.34s	0.32s	0.49s	1.69s	0.27s	0.50s	0.91s	-	0.08s
Prediction	4.11s	0.24s	0.24s	0.42s	1.46s	0.10s	0.97s	-	0.61s	0.08s
Pred.+Opt.	2.72s	0.17s	0.16s	0.44s	0.69s	-	0.48s	0.63s	0.08s	0.07s
User: Q1d	total	initp	endp	shortp	midp	bigp	key	click	tab	mixed
Unassisted	3.42s	0.71s	0.09s	0.27s	0.56s	-	1.79s	-	-	-
Postedit	3.10s	0.81s	0.23s	0.14s	1.09s	-	0.83s	-	-	-
Options	4.35s	0.77s	0.15s	0.30s	1.00s	0.33s	1.76s	0.04s	-	0.00s
Prediction	3.83s	0.57s	0.13s	0.37s	0.72s	-	2.03s	-	-	-
Pred.+Opt.	3.71s	0.55s	0.15s	0.40s	1.10s	-	1.18s	0.32s	-	0.03s
User: Q1e	total	initp	endp	shortp	midp	bigp	key	click	tab	mixed
Unassisted	2.84s	0.28s	0.17s	0.16s	0.32s	0.06s	1.86s	-	-	-
Postedit	2.62s	0.39s	0.25s	0.16s	0.97s	0.12s	0.72s	-	-	-
Options	3.49s	0.14s	0.26s	0.36s	0.56s	0.21s	1.72s	0.21s	-	0.03s
Prediction	2.79s	0.10s	0.32s	0.31s	0.31s	-	1.38s	-	0.30s	0.06s
Pred.+Opt.	3.01s	0.13s	0.30s	0.18s	0.47s	-	1.94s	-	-	-

information is given as seconds per input word (meaning that the total time spent on each activity is divided by the total number of words in the input documents).

Let us take a closer look at two translators: L2b and L1e. L2b is the slowest and a worse than average translator when unassisted. She makes good use of both types of assistance, spending 0.38 seconds on clicking, 0.41 seconds on tabbing (accepting predictions), and using both (0.47 seconds, 0.24 seconds, respectively), when both are offered. This cuts down the time spent on regular typing by 0.9–1.4 seconds. Also, much less time is spend on pauses of various types.

L1e is one of the best translators, but gets hardly any gains from the assistance. The table reveals why: She hardly uses clicks and tabs when offered, and not at all when both are offered. The time spent on typing changes hardly. Nevertheless, she is faster in postediting, mostly due to spending a second less on typing, although some of those gains are eaten up by more pausing, mostly medium pauses.

5.3. Origin of Characters

Time spent on activities is one way to measure the utilization of assistance. Another is to trace back the origin of the characters in the final translation to their generating action. We follow the construction of the translation and record how each character is generated.

Table IV gives a breakdown for each translator for each type of assistance. The break-down into different origins mirrors the time spent on the activities. For instance, translator L2b spent 0.89s, 0.47s, and 0.24s on typing, clicking and tabbing (0.04s on mixed activities — no translator spends significant time on this). The resulting translations contain characters that originate 21%, 44%, and 33%, respectively, from these activities. These numbers do suggest that clicking and tabbing is more efficient in generating characters in the translation, but we have to be careful and also consider the impact on pauses (see next section).

It is interesting to see how many characters are unchanged in postediting. This varies from 74–91% for the different translators. L1e, who has the best performance when postediting, leaves 79% of the characters of the machine translation in place.

5.4. Analysis of Pauses

One important question that we are trying to answer is: What do translators spend their time on? This has consequences for the design of a translation aid, since we want to alleviate the most time-consuming aspects of the translation process to increase its productivity.

Table IV. **Origin of Characters.**

For each character in the final translation, we trace back its origin, which is either a keystroke, a click on an option, a TAB key stroke to accept an prediction, or the MT output as starting point for edits.

	User: L2a				**User: L1a**			
	key	click	tab	mt	key	click	tab	mt
Postedit	9%	-	-	90%	11%	-	-	88%
Options	13%	86%	-	-	8%	91%	-	-
Prediction	10%	-	88%	-	17%	-	82%	-
Pred.+Opt.	21%	31%	46%	-	15%	10%	74%	-

	User: L2b				**User: L1b**			
	key	click	tab	mt	key	click	tab	mt
Postedit	18%	-	-	81%	17%	-	-	82%
Options	59%	40%	-	-	36%	63%	-	-
Prediction	14%	-	85%	-	100%	-	-	-
Pred.+Opt.	21%	44%	33%	-	10%	89%	-	-

	User: L2c				**User: L1c**			
	key	click	tab	mt	key	click	tab	mt
Postedit	18%	-	-	81%	13%	-	-	86%
Options	43%	56%	-	-	14%	85%	-	-
Prediction	45%	-	54%	-	17%	-	82%	-
Pred.+Opt.	30%	68%	1%	-	14%	71%	13%	-

	User: L2d				**User: L1d**			
	key	click	tab	mt	key	click	tab	mt
Postedit	14%	-	-	85%	26%	-	-	73%
Options	99%	0%	-	-	93%	5%	-	-
Prediction	22%	-	77%	-	100%	-	-	-
Pred.+Opt.	15%	0%	84%	-	59%	40%	-	-

	User: L2e				**User: L1e**			
	key	click	tab	mt	key	click	tab	mt
Postedit	17%	-	-	82%	20%	-	-	79%
Options	70%	29%	-	-	77%	22%	-	-
Prediction	32%	-	67%	-	61%	-	38%	-
Pred.+Opt.	73%	4%	22%	-	100%	-	-	-

We already included pauses in the analysis above. But strictly speaking, when examining the log of a translator's actions, all we see are pauses interrupted by actions — key strokes and mouse clicks — that take no measurable amount of time. The length of these pauses reveals valuable information about the cognitive processes of the translator (Schilperoord, 1996).

Recall that we categorize pauses into four categories: Pauses of less than 2 seconds are considered part of a sequence of actions, e.g., the time between key strokes when typing a word. Short pauses of 2–6 seconds indicate some hesitation. Medium size pauses of 6–60 seconds indicate that the translator is thinking and planning her next actions, maybe reading source words or reconsidering some of the already produced output. Longer pauses indicate that the translator is stuck and is trying to solve a difficult translation problem.

However, the thresholds of 2, 6, and 60 are arbitrary and have no more basis than an intuitive understanding of the translation process. Pauses may be of any length. Instead of classifying pauses into arbitrary categories, we may want to look at the whole range of pauses.

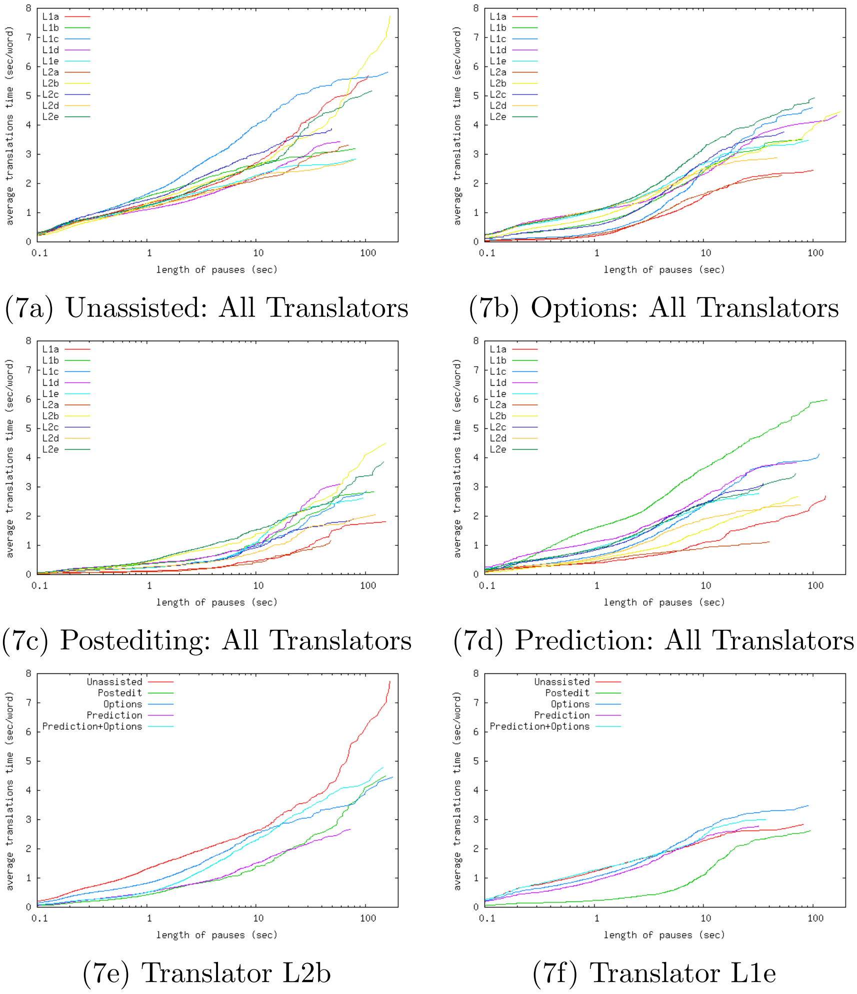
See Figure 7 for an analysis of the pauses of our translators when translating without assistance. Recall that user actions according to our log take no time at all (they happen at specific points in time), and all the time is consumed by pauses between actions. The figure plots on the x-axis the length of pauses and on the y-axis the sum of time spent in pauses of up to that length.

Definition: Accumulated Pause Time. If P is the set of all pauses p in the translation log and $l : p \rightarrow t$ is the function that maps each pause p to its length in seconds t, then the figure shows the graphs of the function

$$acc(t) = \frac{1}{Z} \sum_{p \in P, l(p) \leq t} l(p) \tag{2}$$

Z is the normalization so that $acc(\infty)$ corresponds to the total translation time per input word that we use in all our other tables. Formally the pauses P are generated when translating a set of input sentences S, and each $s \in S$ has a length of $w(s)$. So, $Z = \sum_s w(s)$.

Consider Figure 7a. According to the graph, all translators spend a similar short amount of time in pauses of less than 1 second. Then, the translators diverge. The slowest translator L2b spends about half of her time in pauses of more than 30s. Contrast this to the second slowest translator L1e who spends roughly three quarters of her time in pauses between 3–20s. The fastest translator L1e spends hardly any time pausing more than 20s.



(7a) Unassisted: All Translators

(7b) Options: All Translators

(7c) Postediting: All Translators

(7d) Prediction: All Translators

(7e) Translator L2b

(7f) Translator L1e

Figure 7. **Analysis of Pauses**. Translation time spent on pauses up to a certain length.

This difference in pauses reflects the strikingly different behavior of the translators. As mentioned above, the different lengths of pauses indicate different problems the translators are dealing with. We do not yet feel equipped to further qualify the behavior of translators. We are more concerned with the effect the assistance of the tool has on the translation process.

The Figures 7b–d show the pause graphs for options, postediting and prediction. In all cases pauses of less than one second take up much less time, which indicates that these types of pauses are part of the mechanics of typing. Note that when using options (Figures 7b), there is a steep bump by pauses of length 2–10 seconds. This seems to correspond to the time it takes to visually explore new options, choosing one, and moving the mouse to it. When postediting (Figures 7c) there is very little time spent on pauses shorter than 10 seconds, which indicates that most of the time is spent on contemplating changes, but very little on executing them.

Figure 7e shows the graphs for the weak translator L2b under all five different types of assistance. The graph clearly shows that long pauses during unassisted translation are greatly reduced with assistance. The maximum length of pauses is shortest with the prediction. Otherwise the curves seem similar.

Figure 7f shows the graph for the strong translator L1e, whose curves, except for postediting, are almost identical — another indicator that the assistance is not used. When postediting, most of her time is taken up by pauses of about 7–20 seconds. We can only speculate about the translator's behaviour during such pauses, but intuitively it seems that she is reading more of the machine translation output and looking for mistakes to be corrected.

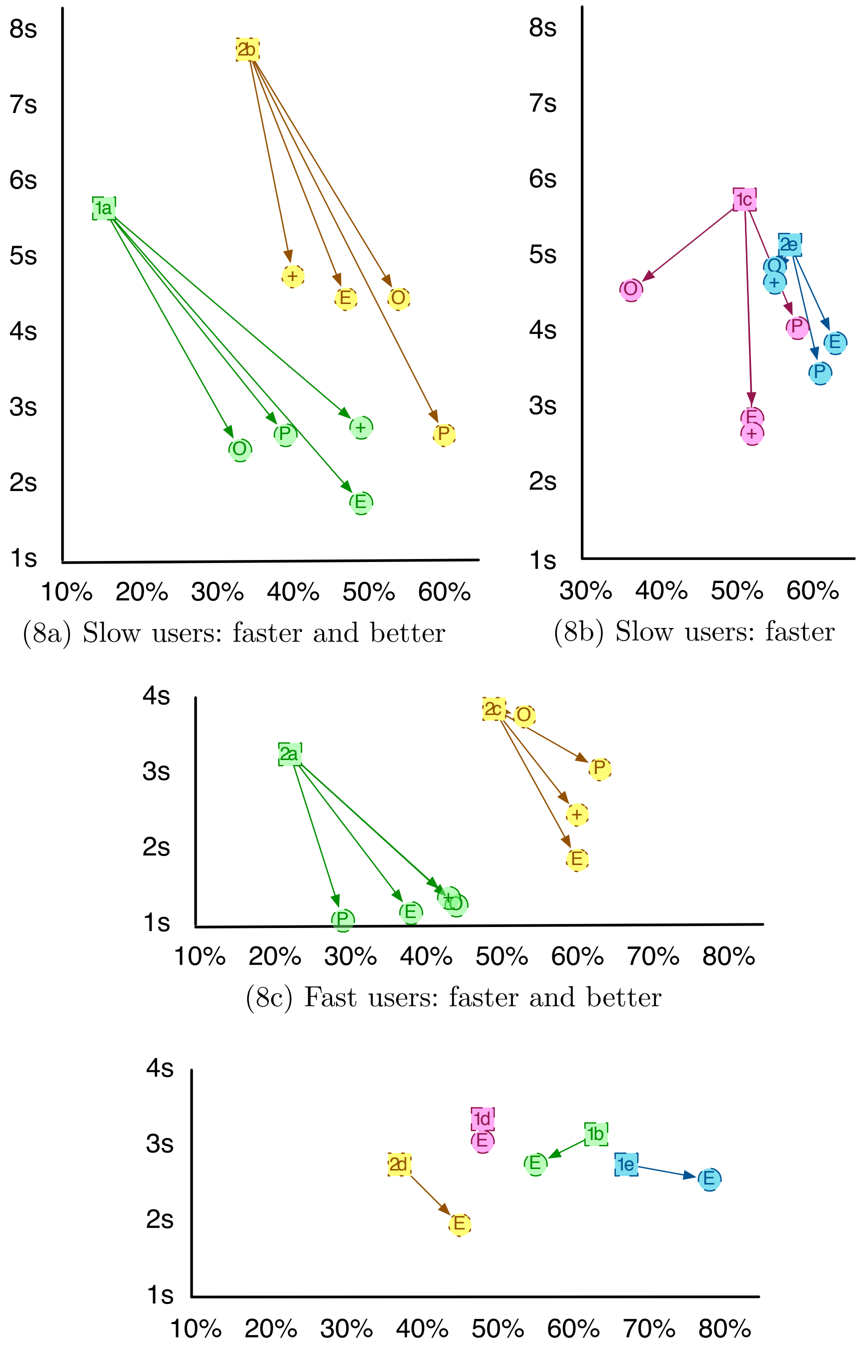
5.5. User Profiles

The different translators have different backgrounds and utilize Caitra differently. We can broadly classify them into three groups (See also Figure 8 for a graphical display):

Slow Translators The four translators L2b, L2e, L1a, and L1c need more than 5 seconds per input word when unassisted. Half of them (L2b, L1a) are very bad unassisted (35%, 16% correct) and become much faster and better, reaching roughly average performance with the assistance (41–61%, 18–34%). The other two (L2e, L1c) are average quality and become faster, but not much better.

Fast Translators The two fast translators L2a and L2c use the assistance offered to them and become even faster with it. However, L2a is a very bad translator (23%), becomes better with assistance, but still below average (30–45%). L2c is an average translator (50%)and becomes slightly better (54–61%).

Refuseniks The four translators L2d, L1b, L1d, and L1e use the assistance sparingly or not at all, and see generally no gains. The two best translators (L1e and L1b) are in this group. Note that the best translator (L1e, 68% correct unassisted) still becomes much better (the record 79%) when postediting.



(8a) Slow users: faster and better

(8b) Slow users: faster

(8c) Fast users: faster and better

(8d) Refuseniks: some improved with post-editing

Figure 8. **User Profiles**. User may be grouped into three classes: slow users (8a,b), fast users (8c), and refuseniks that did not utilize the tool and did not improve (8d). Graphs point from unassisted performance (square with user id) to post-editing (E), Options (O), Prediction (P) and Prediction+Options (+).

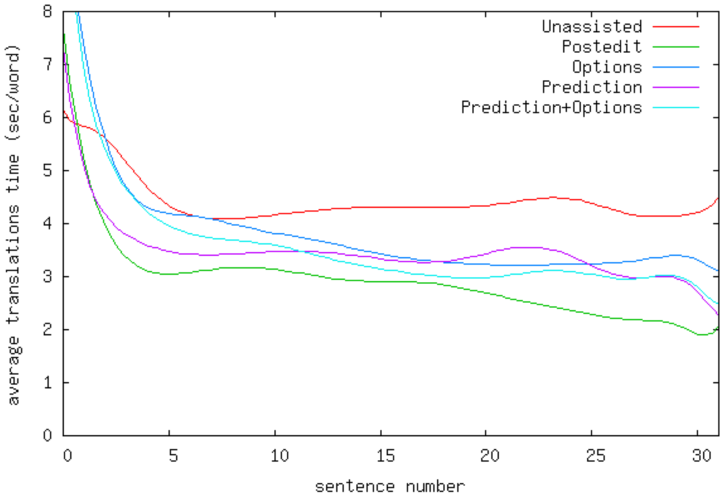


Figure 9. **Learning Curve**. Translators speed up as they translate more sentences and become more experienced with the tool. The graph plots a smoothed curve of the average translation time for each sentence for all translators. For Options and Prediction+Option, the average time for the first sentence was about 10 seconds/word.

5.6. Learning Effect

In our study, the tool in all of its aspects is utilized by novice translators unfamiliar with the types of assistance that we offered to them. While the assistance offered is very intuitive to use, nevertheless the translators may become more proficient with experience.

Translators spent about one hour on 32–46 sentences with each type of assistance. Is there a noticeable learning effect while they become more familiar with the task? See Figure 9 for a graph that plots a learning curve for each type of assistance.

For each translator, we ordered the sentences in the sequence in which they were tackled, and measured the translation time for each sentence. For each sequential number, we computed the average time for all translators. The graph shows a smoothed Bezier curve. Note that we cut off the graph at sentence number 32, since not all blocks had more than 32 sentences.

The new types of assistance that we offered to the translators resulted in very slow performance in the initial sentences, but by sentence number 5, they learned how to use them. From then on, they managed to speed up slightly for the remainder of the task. The speed-up is most pronounced for post-editing, which translators are able to perform in half the time compared to unassisted translation at the end of the task

— while still improving even at that point in time. In contrast, there is no gain in unassisted translation after a start-up bump.

5.7. User Feedback

We requested the translators to fill out a questionary after they completed their translation tasks, and seven did so in time. We ask two multiple choice questions: *Which of the five conditions did you enjoy the most?* Allowing for multiple answers, unassisted was chosen once, postediting once, options twice, prediction twice, and prediction+options three times.

In which of the five conditions did you think you were most accurate? Postediting was chosen once, predictions was chosen once, options was chosen twice, and predictions+options was chosen five times. This self-assessment of quality mostly did not match the human judgement, but it was not completely off the mark either.

We also asked the translators to rank the different types of assistance on a scale from 1 to 5, where 1 indicates *not at at all* and 5 indicates *very helpful.* Postediting received an average rating of 2.9, options a rating of 3.7, prediction a rating of 3.9, and prediction+options a rating of 4.6.

It is striking that postediting was ranked so low, not only in terms of enjoyment, but also in subjective usefulness, while it proved to be as productive as the other types of assistance.

When asked for suggestions for improving the tool, the translators focused on interface issues such as a too small font, being able to finish the translation without clicking the submit button, be able to insert translation options at the cursor position and not just appending them at the end, as well as including a spell checker and grammar checker. Some noted that the options are often wrong and confusing, especially when it comes to prepositions. Some noted that it makes the same mistakes over and over again, and should be able to learn from the corrections.

6. Conclusions and Outlook

We described previously proposed and novel types of assistance for human translators and compared them. The study of the human translation process has shown that assistance improves both speed and accuracy. On average, translators were faster by 16% when given translation options, by 27% when given predictions, by 25% when using both, and by 39% when post-editing. Some translators cut their translation time by more than half.

Users spend similar time on typing. They differ on pauses, especially long pauses. Assistance was most effective with eliminating long pauses of slow translators. Users are also faster with text production by clicking and tabbing — they spend less time on non-pause activities and the ratio of time spent to characters produced is lower for clicking and tabbing (accepting predictions).

There are several aspects that warrant further investigation. Why, when using options, do users spend so much time in pauses of length 2–10 seconds? This may be also due to the mechanics of the tool that may require users to scroll the window to view all options that could be addressed with a better user interface.

Further study of the cognitive processes of translation are needed both to gain insight into what the most time-consuming translation processes are and how they can be alleviated. We would like to investigate what translation problems are the most time consuming, e.g., lexical selection or syntactic restructuring. We are also interested how the tool aids novice vis-a-vis more experienced translators. We would like to expand this scale of qualifications to monolingual speakers of the target language at one end and professional translators at the other end. Professional translators typically handle more technical texts such as product manuals, which we also would like to tackle. Tackling languages with more reordering and lower machine translation quality, such as German–English, is also of great interest to us.

There are many possible extensions and other types of assistance imaginable: Post-editing could be improved by using confidence measures to highlight lower-quality output. The tool could visualize the word alignment between the translation and the input by highlighting the part of the input that corresponds to the current cursor position.

7. Acknowledgments

This work was supported by the EuroMatrixPlus project funded by the European Commission (7th Framework Programme). Thanks to Josh Schroeder for help with Ruby on Rails and Barry Haddow for help with administering the user study.

References

Albrecht, J., Hwa, R., and Marai, G. E. (2009). Correcting automatic translations through collaborations between mt and monolingual target-language users. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics.*

Barrachina, S., Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., Lagarda, A., Ney, H., Tomás, J., Vidal, E., and Vilar, J.-M. (2009). Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Buchweitz, A. and Alves, F. (2006). Cognitive adaptation in translation. *Letras de Hoje*, 41(2):241–272.

Callison-Burch, C., Koehn, P., Monz, C., and Schroeder, J. (2009). Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece. Association for Computational Linguistics.

Carl, M., Jakobsen, A. L., and Jensen, K. T. H. (2008). Studying human translation behavior with user-activity data. In *NLPCS*, pages 114–123. INSTICC Press.

Foster, G., Langlais, P., and Lapalme, G. (2002). User-friendly text prediction for translators. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 148–155, Philadelphia. Association for Computational Linguistics.

Fraser, J. (1996). The translator investigated: Learning from translation process analysis. *The Translator*, 2(1):65–79.

Galvez, M. and Bhansali, S. (2009). Translating the world's information with google translator toolkit.

Jääskeläinen, R. (2001). Think-aloud protocols. In *Routeledge Encyclopedia of Translation Studies*, pages 269–273. Routeledge.

Jakobsen, A. L. (2003). Effect of think aloud on translation speed, revision and segmentation. In Alves, F., editor, *Triangulating Translation*, pages 69–96.

Jakobsen, A. L. and Schou, L. (1999). *Translog Documentation*, volume 24 of *Copenhagen Studies in Language*. Samfundslitteratur.

Jensen, A. and Jakobsen, A. L. (2000). *Translating under time pressure — an empirical investigation of problem-solving activity and translation strategies by non-professional and professional translators*, pages 105–116. Benjamins.

Koehn, P. (2009). A web-based interactive computer aided translation tool. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*.

Koehn, P. and Haddow, B. (2009). Edinburgh's submission to all tracks of the WMT 2009 shared task with reordering and speed improvements to Moses. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 160–164, Athens, Greece. Association for Computational Linguistics.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C. J., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *ACL Demo and Poster Session*.

Kumaran, A., Saravanan, K., and Maurice, S. (2008). wikiBABEL; community creation of multilingual data. In *Babel Wiki Workshop 2008: Cross-Language Communication*.

Langlais, P., Foster, G., and Lapalme, G. (2000a). Transtype: a computer-aided translation typing system. In *Proceedings of the ANLP-NAACL 2000 Workshop on Embedded Machine Translation Systems*.

Langlais, P., Sauvé, S., Foster, G., Macklovitch, E., and Lapalme, G. (2000b). Evaluation of transtype, a computer-aided translation typing system: A comparison of a theoretical and a user-oriented evaluation procedures. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Lörscher, W. (2005). The translation process: Method and problems of its investigation. *Meta*, 50:597–608.

O'Brien, S. (2005). Methodologies for measuring the correlations between post-editing effort and machine translatability. *Machine Translation*, 19(1).

Och, F. J., Zens, R., and Ney, H. (2003). Efficient search for interactive statistical machine translation. In *Proceedings of Meeting of the European Chapter of the Association of Computational Linguistics (EACL).*

Raymond, S. (2007). *Ajax on Rails.* O'Reilly.

Schilperoord, J. (1996). *It's about Time. Temporal Aspects of Cognitive Processes in Text Production.* Rodopi.

Sharmin, S., Špakov, O., Räihä, K.-J., and Jakobsen, A. L. (2008). Effects of time pressure and text complexity on translators' fixations. In *Proceedings of the Symposium on Eye Tracking Research and Applications.*