Predictive Translation Memory: A Mixed-Initiative System for Human Language Translation

Spence Green*, Jason Chuang†, Jeffrey Heer†, and Christopher D. Manning*
*Computer Science Department
Stanford University
{spenceg, manning}@stanford.edu
†Computer Science Department
University of Washington
{jcchuang, jheer}@uw.edu

ABSTRACT
The standard approach to computer-aided language translation is post-editing: a machine generates a single translation that a human translator corrects. Recent studies have shown this simple technique to be surprisingly effective, yet it underutilizes the complementary strengths of precision-oriented and recall-oriented machines. We present Predictive Translation Memory, an interactive, mixed-initiative system for human language translation. Translators build translations incrementally by considering machine suggestions that update according to the user’s current partial translation. In a large-scale study, we find that professional translators are slightly slower in the interactive mode yet produce slightly higher quality translations despite significant prior experience with the baseline post-editing condition. Our analysis identifies significant predictors of time and quality, and also characterizes interactive aid usage. Subjects entered over 99% of characters via interactive aids, a significantly higher fraction than that shown in previous work.

Author Keywords
Language translation; interface design; mixed-initiative; empirical study.

ACM Classification Keywords
H.5.2 Information Interfaces: User Interfaces; I.2.7 Natural Language Processing: Machine Translation

Language translation has all the makings of a mixed-initiative task [11]. Some translations are straightforward and can be routinized while others require linguistic and world knowledge that is difficult to represent. Consider the French word interprète, which can mean ‘interpreter’, ‘artist’, ‘performer’, ‘spokesperson’, or even the pejorative ‘mouthpiece.’ Whether one is a spokesperson or a mouthpiece depends greatly on context. Recall-oriented machines can instantly generate all of these translations, but humans, equipped with world knowledge, may be needed to select the appropriate one. Interactive machine translation—in which humans and machines collaborate—has thus intrigued the research community for decades [8], yet has largely failed in user studies. We hypothesize that classic traps in mixed-initiative design [24], in addition to machine translation (MT) quality, are to blame and are partially responsible for slow commercial uptake.

We present Predictive Translation Memory (PTM), an interactive, mixed-initiative system for language translation. Translation memory is a standard term that refers to a set of bilingual string-string mappings usually consulted via text queries. Our system can be seen as an intelligent translation memory that interactively suggests translations based on user activity. The interface provides source (input language) term lookups, local target (output language) suggestions at the point of text entry (Figure 1), and full translation suggestions to support gisting of meaning. All suggestions update in real-time according to the user-specified partial translation, yet this updating is discreet to minimize distractions. We focus on the interface design, which minimizes gaze shift and maximizes legibility by interleaving source and target text. In contrast, nearly all translator workbenches use a two-column format, much like a spreadsheet. Qualitative feedback from users supports our design choices.

If a principal problem in the design of interactive knowledge-based systems is the transfer of expertise from human to machine [46], then the system should also enable adaptive MT updating, or human-assisted machine translation [44]. Because PTM observes user behavior, the machine is able to refine its suggestions in real-time. Contrast this model with post-editing where the MT system has just one opportunity to produce a suggestion. Our analysis shows that PTM leads to final translations that are significantly different from the initial MT suggestion, but have higher quality according to automatic quality metrics. Crucially, the last machine suggestion is both of high quality and relatively close to the final user translation. This by-product should be useful for future work on automatic MT model updating.
To test the system we conducted the largest published interactive MT user study to date. We hired 32 professional French-English and English-German translators, all of whom were regular users of existing computer-aided translation (CAT) tools. We compared our system to post-editing, which is a strong baseline [29, 21], and is also the most common commercial use of MT. We investigated three questions: (1) Is PTM faster than post-editing? (2) Does PTM enable higher quality translation relative to standard translation quality metrics? (3) Which interactive aids are most effective? We find that while users are slightly slower in the interactive mode—they must read suggestions in addition to translating—they produce higher quality translations. Translators also use the suggestions to a far greater degree than was observed in the largest previous study of interactive MT [37]. Qualitative feedback shows that most users believe that they would be more productive in the interactive mode with practice.

RELATED WORK
The idea of a “human-machine” partnership for language translation—a mixed-initiative design—was proposed as early as 1960 [4]. Interactive machine translation was first investigated in the 1970s [8] as research funding for fully automatic MT, which was deemed infeasible, was discontinued [44]. Here we review both theorized and implemented systems in both the NLP and HCI literature. We also describe how collaborative translation—recently investigated in HCI—can be seen as interactive translation.

Theorized Interactive MT Systems
Bisbey and Kay [8] proposed a system in which pre-editors would annotate the input with linguistic and semantic information, and then target-language post-editors would select from among ranked machine translations. Although it was never implemented, this system became a template for most subsequent work on interactive MT.

In a survey of qualitative studies, Church and Hovy [14] concluded that users regarded post-editing as “an extremely boring, tedious, and unrewarding chore.” They proposed a “superfast typewriter” with an autocomplete key that could fill in the remainder of a word or phrase. Our system draws heavily on their idea of interactive MT as target-text completion.

Evaluated Interactive MT Systems
Early interactive MT systems focused on source pre-editing rather than target generation. Loh and Kong [35] presented a Chinese-to-English system in which human translators annotate the input extensively (phrase boundaries, word senses, etc.). Unpublished results showed greatly reduced post-editing effort to achieve human quality [44]. Whitelock et al. [46] evaluated an English-to-Japanese system in which the machine would query human users about linguistic properties of the English input.

To our knowledge, TransType was the first interactive system [17] that incorporated a modern, statistical MT backend. TransType eschewed source pre-editing in favor of target-text generation aids. The basic unit of translation was the character, whereas our system translates at the word level (however, it provides character-level completions via string-matching in the interface). The TransType UI [18] included an autocomplete dropdown with variable length suggestions selected by an empirical user preference model [19]. Our system instead uses source syntactic constraints to set the prediction length. Their user study [32] found that translation time increased 17% relative to translation from scratch, and that users often typed translations even when the right suggestion was displayed.

TransType2 [16] added a playback mechanism for reviewing user sessions [38] and the ability to accept a full MT suggestion. Additional user studies [36] showed that translators would often accept a full translation and then edit it rather than progressively working through a translation. Our interface explicitly permits this usage via a hot-key, although we most users preferred the interactive aids.

Caitra [30] also included an autocomplete function, and allowed the user to query translations for individual words and phrases. The system could refine its suggestions, but not in real-time: search graphs were pre-computed offline. A user study [29] showed that interactive assistance offered no improvement in terms of time or quality over simple post-editing. In contrast, our system generates new translations each time the user input changes, fully utilizing the search space.

Casmacat [2] is the successor of Caitra. It shares the same backend MT engine, but has a new UI [1] that supports post-editing, text completion, and term lookup. However, the interface is the standard two-column layout and the full MT suggestion is not always available for gisting, a feature that users have found useful in previous studies [21]. Casmacat still uses pre-computed search graphs. A pilot user study [3] showed a slight improvement in automatic quality relative to post-editing.

The system of Barrachina et al. [6] is exceptional in that it provided interactive post-editing. The MT system proposed a partial suggestion that the user would correct and accept. Then the system would recompute its suggestion and the process would repeat. An analysis of keystroke ratio found a reduction relative to translation from scratch. In contrast, our system recomputes suggestions in real-time and passively tracks what the user is doing; the user can ignore the suggestions.

Collaborative Translation
Collaborative translation can be seen as an alternate mode of interactive assistance, albeit a slow one. Morita and Ishida [40, 41] partitioned a translation job between source pre-editors and target post-editors who iteratively refine a translation. The process is seeded by MT. This design hearkens back to the earliest conceptions of interactive translation [28]. A quality evaluation showed that collaborative translation could improve the raw MT output.

Hu et al. [25, 26] proposed a similar process, but added a richer interface and language-independent annotations for collaboration. Collaborative translations were consistently rated higher than the original MT output. However, this method was very slow, requiring days to post-edit fewer than 100 sentences.
Mixed-Initiative Interaction Principles

We believe that the failure of previous interactive MT systems (in user studies) may result from known pitfalls of mixed-initiative design. For example, consider Horvitz’s [24] principle #2: considering uncertainty about a user’s goals. Most previous systems violate this principle by assuming that users need either source or target aids, but not both, or neither. Early interactive systems assumed that pre-editing (source) was most useful [35, 46], whereas later systems like TransType and that of Barrachina et al. [6] focused on the target, sometimes forcing the user to accept portions of the target before proceeding. PTM conceals most aids until the user initiates them, and even allows the user to drop into basic text-editing mode if desired.

Also relevant is Horvitz’s principle #8: minimizing the cost of poor guesses about action and timing. Later systems like CaiTra expose portions of the MT system such as translation rules and associated scores directly on the interface. Confidence is usually coded with color. However, MT systems almost certainly contain a very different internal representation of the translation process than humans. Human translators may not understand why, for example, MT systems can propose non-grammatical and incorrect translations like avec=⇒ them with with high confidence. The translation model is full of these noisy rules that can be very useful to the machine, but uninterpretable to the human. Our interface applies rules to aggregated \( k \)-best predictions to select human-interpretable, high-confidence suggestions.

The design of PTM draws on additional principles of mixed-initiative design. As a baseline, automatic machine translations follows Horvitz’s principle #1: developing significant value-added automation. PTM users can also select alternate translations from a drop-down menu or simply type the desired target text, both in keeping with principle #5: employing dialog to resolve key uncertainties. Following principle #6: allowing efficient direct invocation and termination, interactive translation aids are easily toggled on and off with the Escape key, and source word lookups are invoked only upon mouse hover of source text. Real-time updates of machine translations in response to user input enact principle #9: providing mechanisms for efficient agent-user collaboration to refine results. Finally, visualizing source coverage of translated words supports principle #11: maintaining working memory of recent interactions.

Predictive Translation Memory

The Predictive Translation Memory system is designed for expert, bilingual translators. Previous studies have shown that professional translators work quickly—-they are paid by source words translated—and are usually touch typists [12]. Therefore, the interface is designed to be very responsive, and to be primarily operated by the keyboard. Most aids can be accessed via typing or one of the two hot keys. The current design focuses on the point of text entry and does not include conventional translator workbench features such as workflow management, spell checking, and text formatting.

The system has three components. The client UI is written in JavaScript and runs entirely in a web browser. The UI communicates via a RESTful API with the web service, which is written in Python and backed by a SQL database. The web service manages translation sessions, serving source documents and recording user actions. The web service also forwards translation requests to the MT service, which is a Java servlet running in a J2EE web server. The MT service runs the open source Phrasal MT system, which we heavily modified to support PTM [20]. All UI events are logged to enable analysis and playback. Any translation session can be loaded from the database and replayed in its entirety on the client UI.

In this section, we focus on the UI design decisions. We applied an iterative design process using paper prototyping, rapid prototyping of the client UI connected to the live MT service, a small-scale pilot study, and finally the large-scale user study described in this paper.

Many UI design decisions required significant backend engineering which, in turn, enabled novel interactions. For example, real-time suggestion updating requires the MT service to generate translations at nearly human typing speed.

UI Overview and Walkthrough

We categorized interactions into three groups: source comprehension, target gisting, and target generation. The following outline summarizes the interactions, which are detailed in the following sections. Although the specific design of each feature is novel, those in bold have, to our knowledge, never appeared in a translation workbench:

1. Source comprehension
   (a) Word lookups
   (b) Source coverage: highlight translated words

2. Target gisting
   (a) Full best translation
   (b) Real-time updating: full translation generation

3. Target generation
   (a) Real-time autocomplete dropdown
   (b) Target reordering
   (c) Insert complete translation

Human and machine translations appear together in the target text box. During prototyping we found that users were very sensitive to updates in the text box. They wanted to edit the machine suggestions using conventional text manipulation (cut/paste, etc.) rather than the autocomplete interactions. To clarify ownership of regions of the textbox, we adopted the following target text convention:

Black text belongs to the human translator and is never modified by the machine. Gray text belongs to the machine and is never modified by the human translator.

Interactions allow the user to accept portions of the gray text, which becomes black. Subsequent tests showed that users learned to trust that black text is inviolate, and that gray text is only accessible through certain interactions.

Suppose Joe Translator wants to translate a document from French to English. He opens the document in PTM and sees the
A 
À équiper le centre de formation Studeo qui est accessible aux personnes à mobilité réduite et dont nous travaillons à la réalisation dans le cadre de l’institut Jedlička, avec l’association Tap, et ça depuis six ans.

B
To equip studeo training centre which is accessible to people with reduced mobility and we work to achieve in the framework of the Institute jedlička, with tap, and been there for six years.

C
Des enseignants se rendent régulièrement auprès des élèves de l’institut Jedličkův et leur proposent des activités qui les intéressent et les amusent.

Teachers regularly visit Jedličků Institute students and offered them activities of interest to them and having fun.

D
les étudiants eux-mêmes n’ont pas les moyens de se rendre à des cours, nous essayons de les aider de cette manière.

The students trying to help themselves cannot be required to attend courses, we are

E
dans le cadre du projet dans un no

Institut Jedlička, nous transférerons ce

In pilot experiments we found that the raw alignments were too noisy to show to users. We thus developed MT rule-level heuristics that filter the alignments returned to the interface.

Target Gisting
The most common use of MT output is *gisting* [31, p.21]. A rough translation is often sufficient to convey meaning. Translators find MT useful as an initial draft [21].

Full Best Translation
The gray text below each black source input shows the best MT system output (Figure 2, B). As Joe works on the focus translation, the gray text adjusts in the target textbox to show the best suggested completion (Figure 2, E).

Real-time Updating
When Joe starts working on a source sentence, the gray text will update to the most probable completion (Figure 2, E) for his partial translation (black text). The update always appears as a gray completion following the black translation prefix. The human and machine refine the translation collaboratively (Horvitz’s principle #9: *providing mechanisms for efficient agent-user collaboration to refine results*) with the machine in a strictly responsive role.

Target Generation
The target textbox shows both the user and machine state simultaneously. This allows Joe to accept parts of the machine suggestion without touching the mouse. The black portion is a text editor: Joe can cut, copy, paste, or otherwise manipulate the black text. However, the gray text is immutable. It cannot be highlighted with the cursor or changed. Joe accesses it through three interactions.
Autocomplete Dropdown
The autocomplete dropdown at the point of text entry is the main translation aid (Figure 2, D). Each time Joe enters a target word or otherwise edits the black prefix, the MT service returns a list of completions conditioned on the accepted prefix. Up to four unique suggestions appear in the target dropdown. The top suggestion can be selected via either the Tab or Enter keys. The dropdown can be navigated with the arrow keys, the mouse, or by beginning to type the desired suggestion. Suggestions that do not match the partial word are filtered until the desired suggestion is at the top of the list. Then the Tab or Enter keys can be used to select it.

The suggestion length is based on the syntax of the source language. As an offline, pre-processing step, we create syntactic parses of the source input with Stanford CoreNLP [39]. The UI combines those parses with word alignments from the full translation suggestions to project syntactic constituents to the target. Syntactic projection is a very old idea that underlies many MT systems (see: [27]). Here we make novel use of it for suggestion prediction filtering. Presently, we project noun phrases, verb phrases (minus the verbal arguments), and propositional phrases. If no constituents can be projected, then the UI backs off to single-word suggestions.

Target Reordering
So far we have assumed a left-to-right generation scheme, but that design fails for long-distance reordering. For example, in English-to-German translation, some verbs will need to be moved to the very end of a sentence. To that end, the UI supports keyboard-based reordering.

Suppose that Joe sees the (partially correct) suggestion Wirtschaftliche Offen“ec’omic offences‘ in the gray text (Figure 4) and wants to move that suggestion to the insertion position. Joe can begin typing that string, and the UI will update the autocomplete dropdown with matching strings from the gray text. Consequently, sometimes the autocomplete dropdown will contain suggestions from several positions in the full suggested translation. The user can insert the suggestion from the dropdown in the usual ways.

Layout and Typographical Design
Carl [12, p.11] showed that translators spend up to 20% of any translation session reading source text and revising target text, and that harder translations can significantly increase this fraction. However, we noticed that most translator workbenches are optimized for typing, and conform to a tabular, two-column spreadsheet layout—source and target are aligned by row. A spreadsheet design may not be optimal for reading text passages.

Our UI is based on a single-column layout so that the text appears as it would in a document. Sentences are offset from one another primarily because current MT systems process input at the sentence-level. We interleave target-text typing boxes with the source input to minimize gaze shift between source and target. Contrast this with a two-column layout in which the source and target focus positions are nearly always separated by the width of a column.

The compact, single-column layout can obscure the boundaries between source and target, especially for languages with similar writing systems. We found that rendering source and target in different typefaces restored legibility. In our UI, source is rendered in a serifed font, which is commonly used for body text [45]. The target text appears in a monospaced, sans-serif font. Monospaced fonts are conventional for text entry forms. We chose the Paratype2 font family, which features a large x-height for more readable type [45].

Summary of MT Service
Statistical MT systems come in two general flavors: phrase-based and hierarchical/syntactic. Phrase-based systems decode input (i.e., search for translations) left-to-right and can run in $O(n)$ time. Hierarchical/syntactic systems are not restricted to left-to-right processing, but decode with the slower $O(n^3)$ CKY parsing algorithm. Although the left-to-right constraint may not necessarily correspond to the human translation process, we found in pilot studies that users tended to value speed and responsiveness, hence we chose a phrase-based system.

---

Figure 3: Source word lookup menu (top), which only appears with the autocomplete dropdown (bottom) when the user hovers over a source token. The word lookup suggestions do not depend on the partial translation Teachers, so the list of suggestions is different from those shown in the autocomplete dropdown for the same term.

Figure 4: Target reordering feature. The user can move a suggestion to the current editing position by typing the prefix. The system predicts the suggestion length.

Insert Complete Translation
At any time, Joe can accept the full completion by pressing the Control+Enter hot key. Notice that if the user presses this hot key immediately, the full suggestion is inserted, and the interface is effectively a post-editor. This feature greatly accelerates translation when the MT is mostly correct, and the user only wants to make a few changes.

Both types of MT systems are trained in a sequential pipeline: word alignment, translation rule extraction, model parameter learning, and finally decoding of source input. A deployed system like ours must also perform pre- and post-processing of inputs and outputs since the MT system typically expects and generates lowercased, tokenized (e.g., punctuation is separated from words) text that should not be shown to the user. Our system is trained offline prior to usage, but performs pre- and post-processing online inside the MT service.

The UI design required considerable backend engineering to support real-time suggestion updating [20]. Here we summarize a few of the more interesting details.

First, to support suggestions that match a user prefix, we implemented a novel variant of forced decoding. Forced decoding constrains an MT system to produce a specific translation and is sometimes used for parameter learning or diagnostics. Our variant is called prefix decoding: we force the system to match the user prefix, and then allow it to translate freely the remainder of the source input. The challenge is that the user prefix may contain words that the system has never seen before, and forced decoding ordinarily fails in this scenario. To solve this problem, we generate synthetic translations from each source word to each unseen target word on-the-fly, and allow the MT system to guess which rules to use.

Second, in pilot experiments we found that unless the MT service could return translations in less than about 300ms, users deemed the UI as “sluggish.” The phrase-based decoding algorithm is an instance of beam search, an approximate procedure that maintains a ranked list of candidates. Reducing the list (beam) size at decoding time increases search speed but usually reduces translation quality, a classic tradeoff. However, we found that if we reduced the beam size during parameter learning, and ran the learning procedure longer, we could mostly recover these losses.

Finally, although we made the MT system considerably faster, it is nonetheless slow relative to conventional AJAX requests (e.g., database queries). Since requests arrive at approximately typing speed while the translator works, the MT service can exhaust its request handling threads waiting on the MT system, and new requests cannot be processed. To solve this problem, we implemented asynchronous request handling via the Java Servlet 3.0 suspend API. Requests can be suspended while waiting for the MT system so that new requests can be queued. This architecture is critical to making the UI responsive.

EXPERIMENTAL DESIGN

We conducted a language translation experiment with a 2 (translation conditions) x n (source sentences) mixed design, where n depended on the language pair (Table 1). Translation conditions (post-edit and PTM/interactive) and source sentences were the independent variables (factors). Experimental subjects saw all factor levels, but not all combinations, since one exposure to a sentence would certainly influence another.

We randomized the assignment of sentences to translation conditions and the order in which the translation conditions appeared to subjects. At most five sentences appeared per screen, and those sentences appeared in the source document order. Subjects received untimed breaks both between translation conditions and after about every five screens within a translation condition.

Subjects completed the experiment remotely on their own hardware. They received personalized login credentials for the web service, which administered the experiment. Upon login, subjects were assured that no identifying personal information would be recorded, and were asked to consent to having translation session information recorded for playback and analysis. Subjects then completed a demographic questionnaire that included information such as prior experience with CAT and self-reported language proficiency. Next, subjects completed a training module that included a 4-minute tutorial video and a practice “sandbox” for developing proficiency with the two translation UIs. Then subjects completed the translation experiment. Subjects could move among sentences within a screen, but could not go back to previous screens to make corrections. Finally, they completed an exit questionnaire. Most of the questions asked users to rate parts of the experiment and the interfaces according to a 5-point Likert scale. Free-form responses to several questions were also solicited.

To minimize the number of learned interactions, we replaced the document navigation hot keys with mouse navigation. To force a contrast with post-edit, we also disabled the Escape key so that subjects could always see at least the full target translation (gray text) and the autocomplete drop-down. Subjects completed the experiment under time pressure. We used an idle timer identical to that of Green et al. [21], and asked subjects to complete the experiment in a single day.

Linguistic Materials

We chose two language pairs: French-English (Fr-En) and English-German (En-De). French and English are typologically similar, whereas English and German can have different canonical word orders. Anecdotally, French-English is a very easy language pair for MT, whereas English-German is very hard due to long-distance reordering and complex German morphology (e.g., case, gender agreement, etc.).

We chose three text genres: software, medical, and informal news. These genres differ significantly from the majority of the data used to train the MT system, thus replicating the domain mismatch commonly occurring in the translation/localization industry. The software data came from the graphical interfaces of Autodesk AutoCAD and Adobe Photoshop. The medical data was a drug review from the European Medicines Agency. These data came from the TAUS data repository3 and contained professional human reference translations. The informal news data came from the Workshop on Machine Translation (WMT) 2013 shared task test set [9].

We expected that the software would be hardest, the medical data would be moderately difficult, and the newswire would be easiest. The exit survey confirmed that the software data was indeed hardest, but that the newswire was more challenging than the medical data. Despite the presence of jargon in the

3http://www.tausdata.org/
We analyze PTM and post-edit in terms of the two sentence-level response variables: time and quality. We also measure interactive aid usage by UI event gross statistics. For quality, we choose BLEU\[34\], which is the sentence-level variant of the corpus-level BLEU metric \[42\]. Both variants are computed relative to a reference translation, and combine measures of string and length similarity. To maximize BLEU, a system must produce a translation that contains similar n-grams and is of similar length to the reference. Values are conventionally reported as pseudo-percentages, with 100 indicating an exact match with the reference. BLEU has numerous well-known limitations like invariance to permutations \[10\]. Nevertheless, it correlates surprisingly well with human judgment \[13\] and is thus the standard in MT research.

More importantly, BLEU is an MT-tunable metric. Horvitz's principle \#12 is continuing to learn by observing: a true mixed-initiative MT system will improve with use. Human assessment is the final arbiter when evaluating MT systems, yet it is slow and expensive, preventing its practical application for tuning to human feedback. Therefore we focus on automatic quality assessment, leaving a human evaluation to future work.

We excluded one Fr-En subject and two En-De subjects from the models. One subject misunderstood the instructions of the experiment and proceeded without clarification; another skipped the training module entirely. The third subject encountered a technical problem that prevented session logging.

**TIME RESULTS AND ANALYSIS**

Our analysis uses linear mixed effects models (LMEM) built with the lme4 \[7\] R package. LMEMs are more robust to type II errors than ANOVA when factors represent samples from larger populations. In our case, both subjects and source sentences are small samples from the human and linguistic populations, respectively.

The log of time (in seconds) is the response and the independent variable of interest is translation condition. We also found several other significant covariates and added them to the model. The maximal random effects structure \[5\] includes random intercepts and slopes for subject, source sentence, and text genre.

Table 2 shows the results. PTM is slightly slower for both language pairs. For Fr-En, the LMEM predicts a mean time (intercept) of 46.0 sec/sentence in post-edit vs. 54.6 sec/sentence in PTM, or 18.7% slower. For En-De, the mean is 51.8 sec/sentence vs. 63.3 sec/sentence in PTM, or 22.1% slower.

The other significant effects reveal more about translator behavior and differences between the two language pairs. Translators were consistently slower for longer sentences (log source length) and when suggestions required more editing (normalized edit distance). Females were slower, but only at a statistically significant level in En-De. The unbalanced En-De subject pool (Table 1) may be the cause.

The significance and coefficient of ui order shows that subjects improved in both conditions with practice. Subjects were significantly slower in En-De, but there is also a significant interaction between interface condition and ui order, meaning that subjects were significantly faster in PTM as the experiment progressed. Figure 5 shows visual evidence.

The high significance level of no edit shows that accurate initial MT provided significant acceleration.

**Qualitative Time Analysis**

The time models show that users were initially slower with PTM, but that they improved over the course of the session. Many users believed that with more practice they could translate faster with PTM. However, this optimism came with the

---

**Table 1: Full user study summary. We also conducted a pilot study with four professional Fr-En translators that cost $981.52.**

<table>
<thead>
<tr>
<th></th>
<th>Fr-En</th>
<th>En-De</th>
</tr>
</thead>
<tbody>
<tr>
<td>#subjects</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>male/female</td>
<td>7/9</td>
<td>4/12</td>
</tr>
<tr>
<td>#source tokens</td>
<td>3,003</td>
<td>3,002</td>
</tr>
<tr>
<td>#source sentences</td>
<td>150</td>
<td>173</td>
</tr>
<tr>
<td>$ / subject</td>
<td>$265.26</td>
<td>$265.18</td>
</tr>
<tr>
<td>Total</td>
<td>$4,244.16</td>
<td>$4,242.88</td>
</tr>
<tr>
<td>Grand Total</td>
<td>$8,487.04</td>
<td></td>
</tr>
</tbody>
</table>

---

\[4\] http://www.proz.com

\[5\] We ended up excluding the noisy CommonCrawl Fr-En data.
When asked, “In which interface did you feel most productive?”, the proportions were the same: all but two subjects chose the same interface for both questions. The slight preference for post-edit may result from prior familiarity with that mode. When asked to respond to the statement, “I would use interactive translation features if they were integrated into a CAT product”, 11 subjects chose “Strongly Agree” and nine responded “Agree”; only four disagreed with the statement. More encouragingly, when presented with the statement, “I got better at using the interactive interface with practice/experience,” 25 subjects agreed or strongly agreed, and none of the subjects disagreed. Free-form responses elaborated on this theme:

The post-edit mode was easier at first, but in the end the interactive mode was better once I got used to it.

I felt that if I had time to use the interactive tool and grow accustomed to its way of functioning, it would be quite useful...

I am used to this [post-edit], this is how Trados [the pre-eminent CAT tool] works.

**TRANSLATION QUALITY RESULTS AND ANALYSIS**

We build LMEMs with the same random effects structure but with the log of BLEU+1 as the dependent variable. Table 3 shows the results. For Fr-En, the LMEM predicts a mean (intercept) BLEU+1 score of 33.7 for post-edit and 34.6 for PTM. For En-De, the mean is 25.4 for post-edit and 26.3 for PTM. For both language pairs there is a significant main effect for interface condition.

The inclusion and significance of log time—the dependent variable in the previous section—merits discussion. We hy-
We have two options: a multivariate model for time and quality, we do not know the conditions under which the independent variables work. In post-edit subjects tended to use the initial MT suggestions. BLEU is a measure of similarity with the independently generated references less useful for general CAT evaluation. A human quality assessment between PTM and post-edit is needed for a final verdict.

**Qualitative Quality Analysis**

Subjects perceived our baseline MT systems to be unusually effective. They often submitted lightly edited translations in the post-edit condition. The baseline MT systems were trained on a small amount of in-domain TAUS data, which probably increased accuracy relative to a generic MT system. This may have benefitted the post-edit condition more than PTM:

I found the machine translations (texts in gray) were of a much better quality than texts generated by Google Translate

The translations generally did not need too much editing, which is not always the case with machine translations.

Some users articulated aesthetic critiques about MT in general. MT systems tend to produce more literal translations. When users wanted to render more stylistic translations, they believed that PTM was less useful:

...choosing a very different translation approach (choice of words, idioms with no equivalent in English...) would be like going against the current—but may have provided a better quality.

...distracts from own original translation process by putting words in head that confuse [my] initial translation vision

...the translator is less susceptible to be creative

Some users noticed and seemed to resist priming by MT suggestions, even if priming can lead to better translations [21].

**INTERACTIVE AID RESULTS AND ANALYSIS**

We analyzed the methods subjects used to enter text by aggregating UI events into five modes of target generation: autocorrect-best, source suggestion, autocorrect-alternative, interactive typing, and non-interactive typing.

Autocorrect-best refers to users accepting the best machine translation, turning a block of gray text to black either incrementally (via tabbing) or completely (via the Insert Complete Translation interaction). Source suggestion refers to users looking up the translation of a source word, and inserting it into the text box via a mouse click. Autocorrect-alternative refers

---

**Table 3:** LMEM sentence-level quality (BLEU +1) results for each fixed effect with contrast conditions for binary predictors in (1).

<table>
<thead>
<tr>
<th></th>
<th>Fr-En</th>
<th>En-De</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>log time</td>
<td>gender (Female)</td>
</tr>
<tr>
<td><strong>p</strong></td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td><strong>sign</strong></td>
<td>+</td>
<td>−</td>
</tr>
</tbody>
</table>

---

**Table 4:** Corpus-level quality for the two translation conditions. BLEU is the human translations with respect to the independent references; HBLEU is the initial MT suggestion with respect to the human translations. For both metrics a higher score indicates greater similarity.

<table>
<thead>
<tr>
<th></th>
<th>Fr-En</th>
<th>En-De</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>post-edit</td>
<td>PTM</td>
</tr>
<tr>
<td></td>
<td>BLEU</td>
<td>HBLEU</td>
</tr>
<tr>
<td>Fr-En</td>
<td>38.1</td>
<td>63.7</td>
</tr>
<tr>
<td>En-De</td>
<td>38.4</td>
<td>62.6</td>
</tr>
</tbody>
</table>

---

6Consequently, we remove log source length and normalized edit distance from the quality model.

7Conversely, all humans exceeded the baseline En-De MT system.

8It is not possible to compute statistical significance because the translations in each condition are unbalanced. Recall that we filtered three subjects completely, and also removed individual translations for which the idle timer expired.
We asked the subjects to select the least and most useful interactive aids. Target aids were deemed most useful. The target full translation (gray text) received the most votes (11) followed closely by autocomplete (8). Surprisingly, source aids were deemed least useful, with subjects equally ambivalent about the source coverage aid (11) and the word lookup feature (11).

We also asked subjects to rate each aid on a 5-point Likert scale. Aggregating these ratings leads to a global ranking over aids. Here subjects rated autocomplete highest, the target full translation second, and word lookup third. We also asked subjects to rate the usefulness of the suggestion reordering and length prediction features. The majority of users (20) either agreed or strongly agreed that the length prediction was useful, validating our syntactic projection technique. Subjects were less enthusiastic about reordering, with half disagreeing that it is useful. However, this feature is admittedly the most complex interaction in the UI so it probably takes the longest to learn and master. Additional development might focus on simplifying or improving the reordering feature.

CONCLUSION
We presented Predictive Translation Memory, a new interactive, mixed-initiative language translation system. A large-scale evaluation on two language pairs showed that subjects approach the speed of simple post-editing but with an improvement in automatically evaluated translation quality. The baseline post-edit condition was very strong since all subjects were regular users of post-editing software. Qualitative analysis showed that users liked the interactive aids, and many believed that with more practice, PTM could increase their productivity. Future work should focus on the potential for PTM to improve quality according to a human assessment.

Log analysis revealed that users engaged interactive aids to a greater degree than in previous work on interactive MT. We hope to exploit the rich interaction logs generated by these aids to create an MT system that learns and adapts to each user. The automatic quality results auger well for this research direction. Comparison of French-English and English-German strongly suggested that MT accuracy does affect user behavior. An adaptive system could further increase productivity, especially for language pairs with poor baseline MT.

ACKNOWLEDGEMENTS
We thank TAUS for access to their data repository. The first author is supported by a National Science Foundation Graduate Research Fellowship. This work was also supported by the Defense Advanced Research Projects Agency (DARPA) Broad Operational Language Translation (BOLT) program through IBM. Any opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the view of either DARPA or the US government.

---

**Table 5:** Percentage (%) of editing events corresponding to the five modes of target generation using the PTM system.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Fr-En</th>
<th>En-De</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>autocomplete-best</td>
<td>17.46</td>
<td>7.85</td>
<td>12.03</td>
</tr>
<tr>
<td>interactive typing</td>
<td>45.58</td>
<td>43.06</td>
<td>44.16</td>
</tr>
<tr>
<td>non-interactive typing</td>
<td>36.94</td>
<td>49.06</td>
<td>43.79</td>
</tr>
<tr>
<td>source suggestion</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>autocomplete-alternative</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
</tbody>
</table>

**Table 6:** Percentage (%) of text entered (measured by the number of characters modified) via the five PTM modes of target generation.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Fr-En</th>
<th>En-De</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>autocomplete-best</td>
<td>71.09</td>
<td>60.46</td>
<td>65.61</td>
</tr>
<tr>
<td>interactive typing</td>
<td>15.92</td>
<td>18.37</td>
<td>17.18</td>
</tr>
<tr>
<td>non-interactive typing</td>
<td>12.90</td>
<td>20.93</td>
<td>17.04</td>
</tr>
<tr>
<td>source suggestion</td>
<td>0.04</td>
<td>0.06</td>
<td>0.05</td>
</tr>
<tr>
<td>autocomplete-alternative</td>
<td>0.05</td>
<td>0.19</td>
<td>0.12</td>
</tr>
</tbody>
</table>