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 humans 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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
UIST 2014, October 5–8, 2014, Honolulu, HI, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3069-5/14/10 ...$15.00.
http://dx.doi.org/10.1145/2642918.2647408

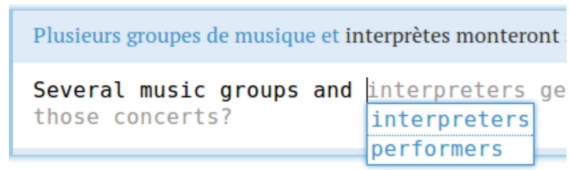

Figure 1: Example of three interactive aids in PTM. The system predicts which French input words have been translated and shades them in blue. The gray text in the typing box shows the best system prediction for the rest of the translation. The user can accept parts of the system suggestion from the dropdown.

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-edit?, (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,[1] 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.

[1]Personal communication with M. Kay.

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, generating 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

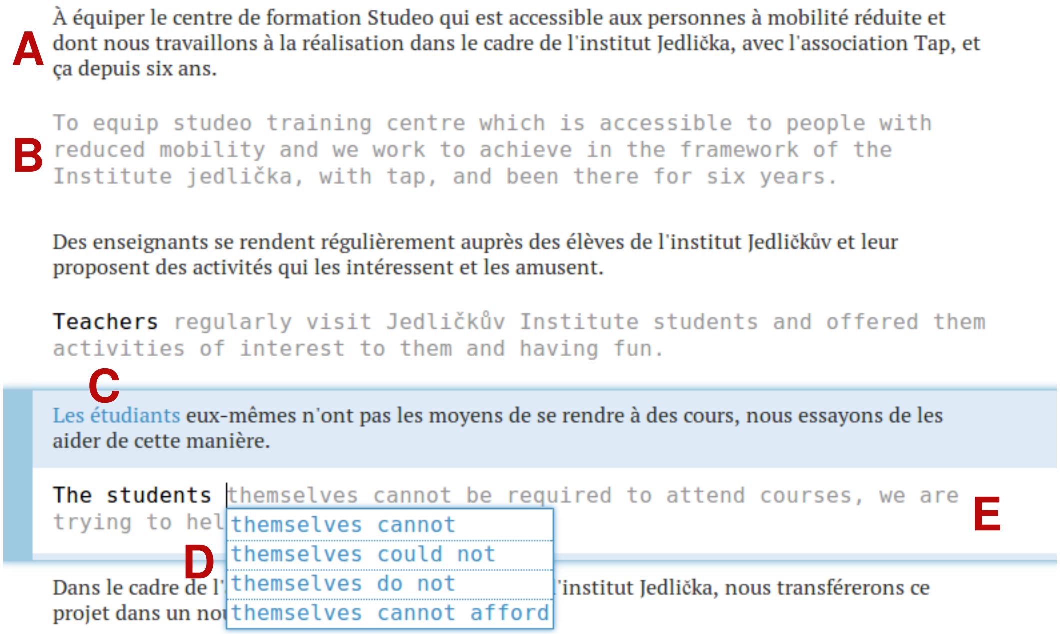

Figure 2: Main translation interface. The interface shows the full document context, with English source inputs (A) interleaved with suggested target translations (B). The sentence in focus is indicated by the blue rectangle, with translated source words shaded (C). The user can navigate between sentences via hot keys. The user can also hide/unhide the autocomplete dropdown (D) and full translation suggestions (E) by toggling the Escape key.

screen in Figure 2. The French sentences (A) are interleaved with English suggested translations (B). Joe must then finalize the translations. When an English translation is finalized, the text becomes black. The following sections describe the interactive aids available to Joe.

Source Comprehension

Word Lookup

Users often trace the source with the mouse cursor while reading [21]. When Joe hovers over source words in the main UI (Figure 2), a menu of up to four ranked translation suggestions appears (Figure 3). We previously proposed [21] showing suggestions proactively for certain parts of speech, but prototyping revealed that this design distracted users. Consequently, we chose a direct-invocation design following Horvitz's principle #6: *allowing efficient direct invocation and termination*. The menu is populated with individual translation rules from the MT translation model. This query does not depend on source context, so it does not require full MT and is very fast, usually returning in under 50ms. The width of the horizontal bars indicates confidence, with the most confident suggestion placed at the bottom, nearest to the cursor. The user can insert a translation suggestion by clicking.

Source Coverage

The interface predicts which source words have already been translated and shades them in blue (Figure 2, C). Joe can quickly find untranslated words in the source. The source coverage is a record of translation interactions consistent with Horvitz's principle #11: *maintaining working memory of recent interactions*. The interaction is based on the word alignments between source and target generated by the MT system. 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.

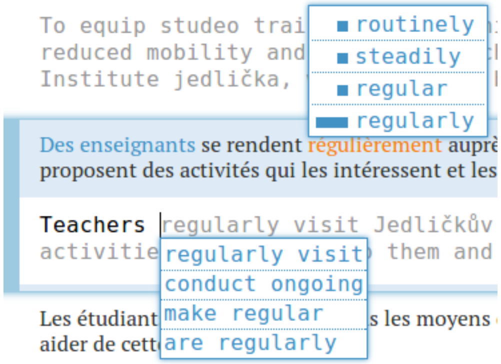

Figure 3: Source word lookup menu (top), which only appears with the autocomplete dropdown (bottom) when the user hovers over a source token. The word lookup suggestions do not depend on the partial translation *Teachers*, so the list of suggestions is different from those shown in the autocomplete dropdown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

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 prepositional 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 Offences* 'economic 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.

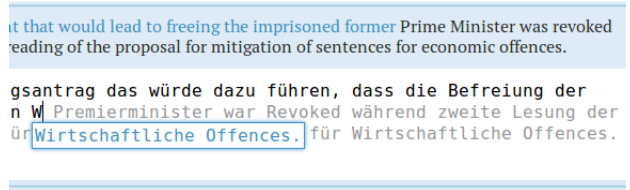

Figure 4: Target reordering feature. The user can move a suggestion to the current editing position by typing the prefix. The system predicts the suggestion length.

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.

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 Paratype[2] 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.

[2] `http://www.paratype.com/public/`

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 repository[3] 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

[3] `http://www.tausdata.org/`

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

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

drug review, the sentences were formulaic, and the translators apparently did not need medical expertise to translate them.

The Fr-En dataset contained 3,003 source tokens; the En-De dataset contained 3,002. Average human translators process about 2,700 source tokens per day [43, p.36], so the experiment was designed to replicate a slightly demanding work day.

Selection of Subjects

We recruited professional, freelance translators on Proz, which is the largest online translation community.[4] We posted ads for both language pairs at $0.085 per source word, an average rate in the industry. In addition, we paid $10 to each translator for completing the training module. Table 1 summarizes the experimental subjects and data.

All subjects had significant prior post-editing experience with commercial CAT workbenches. We tried to balance the subject pool by gender, but could not find enough male participants.

MT System

We trained large-scale Fr-En and En-De translation systems on all of the constrained data from the WMT 2013 shared task.[5] For coverage, we also added 61k parallel segments of TAUS data to the En-De bitext, and 26k TAUS segments to the Fr-En bitext. We aligned the parallel data with the Berkeley aligner [33] and built 5-gram language models with lmplz [23]. The MT model contained 18 baseline features [22]. We set the beam size to 800 for both parameter learning and decoding. The held-out parameter tuning set contained a third medical data, a third software, and a third newswire.

Evaluation Metrics

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+1 [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

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

[5] We ended up excluding the noisy CommonCrawl Fr-En data.

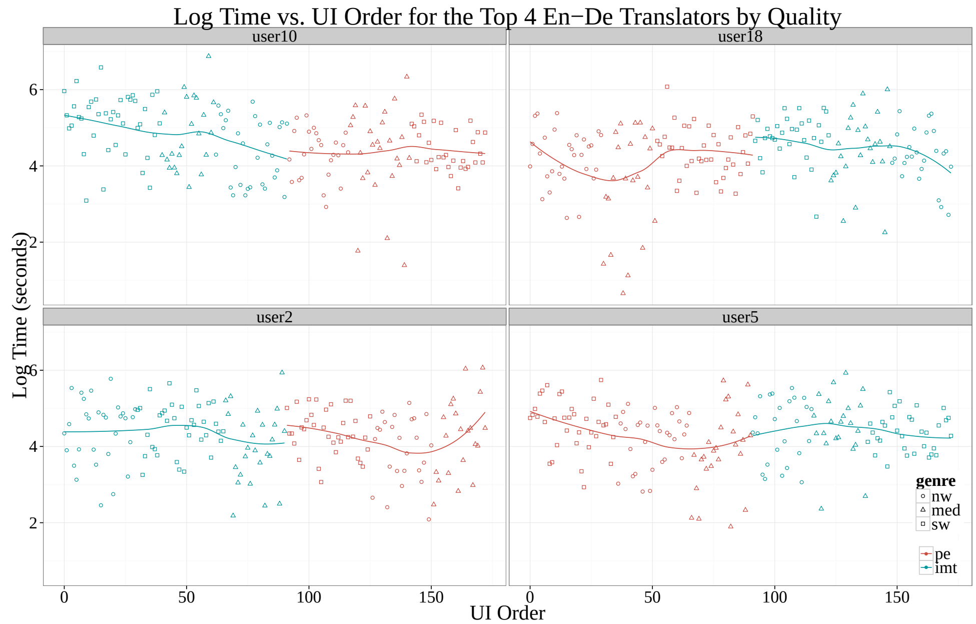

Figure 5: Log time vs. ui order for the top four En-De subjects (according to quality) with loess trend lines. In the post-edit condition (red), three of the four subjects maintain a relatively steady level of performance, whereas in the PTM condition (blue) all four subjects improve with practice. Recall that the order of translation conditions and documents were randomized.

	Fr-En		En-De	
	sign	p	sign	p
ui (PTM)	+	∘	+	●●
ui order	−	●	−	●●
normalized edit distance	+	●●●	+	●●●
no edit (True)	−	●●●	−	●●●
gender (Female)	+		+	●
log source length	+	●●●	+	●●●
ui (PTM) : ui order	+		−	●

Table 2: LMEM time results for each fixed effect with contrast conditions for binary predictors in (). *normalized edit distance* is computed with respect to the initial MT suggestion, and is normalized by the source length. *ui order* is the order of the source sentence in each ui condition for each subject. The sign of the coefficients can be interpreted as in ordinary linear regression. Statistical significance was computed with a likelihood ratio test: ●●● $p < 0.001$; ●● $p < 0.01$; ● $p < 0.05$; ∘ $p < 0.1$.

caveat that the interactive mode was necessarily more labor intensive. There are more aids to operate and more information to read and analyze:

> Because you spend more time on each word, you have opportunity to see alternative translations.

Subjects noticed that MT quality greatly affected the usefulness of the interactive aids:

> If drop-down suggestions are not of a good quality, reading (without selecting them) may consume extra time.

When asked, "In which interface did you feel most productive?", subjects were almost evenly divided, with 15 selecting post-edit and 14 choosing PTM. When asked, "In general, which interface did you prefer?", 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 preeminent 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-

	Fr-En		En-De	
	sign	p	sign	p
ui (PTM)	$+$	∘	$+$	•
no edit (True)	$+$	•••	$+$	∘
gender (Female)	$-$		$+$	
log time	$-$	•••	$-$	•••

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

	Fr-En		En-De	
	BLEU	HBLEU	BLEU	HBLEU
post-edit	38.1	63.7	29.4	44.1
PTM	38.4	62.6	29.5	41.0

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.

pothesize some correlation between time and quality. Consider a subject who simply submits the initial MT immediately without any editing. Absent perfect MT this strategy optimizes time at the expense of quality. The time analysis also showed that translation with PTM tends to be slower than post-edit. We have two options: a multivariate model for time and quality, or inclusion of time as a independent variable.

Here we include time as an independent variable since it also captures an important property of BLEU+1. Time is positively correlated with source length ($\rho = 0.53$ for Fr-En and $\rho = 0.43$ for En-De): longer sentences take longer to translate.[6] It is *negatively correlated* with BLEU+1 ($\rho = -0.21$ for Fr-En and $\rho = -0.24$ for En-De). This is a common property of automatic metrics. Since a longer sentence has many more possibilities for translation, the overlap between any single translation and any single reference tends to decrease with length. The models reflect this tendency: for both language pairs *log time* has a negative coefficient.

The significant predictor *no edit* has a positive coefficient for Fr-En. We found that the baseline Fr-En MT system produced a higher corpus-level BLEU score than *any* human subject.[7] This result corroborates previous work [15][31, p.229] on the inability of BLEU to discriminate among accurate translations.

We also computed corpus-level BLEU and HBLEU (Table 4). BLEU is a measure of similarity with the independently generated references. Overall, users produced slightly higher BLEU scores with PTM.[8] HBLEU is measure of similarity with the initial MT suggestions. In post-edit subjects tended to deviate less from the initial MT than in the interactive mode.

We hypothesize two explanations for the results in Table 4. First, our previous work on unaided vs. post-edit [21] showed that MT suggestions prime translators. PTM exposes translators to many more alternatives, encouraging them to deviate further from the initial MT suggestion (lower HBLEU). Second, we do not know the conditions under which the independent references were generated. For example, the En-De references contain English transliterations or loan words for many medical terms, whereas the subjects in our study tended to seek faithful target-language translations. The automatic metrics are sensitive to lexical differences possibly making independent references less useful for general CAT evaluation. A human quality assessment between PTM and post-edit is needed for a final verdict.

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

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

[8]It 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.

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: autocomplete-best, source suggestion, autocomplete-alternative, interactive typing, and non-interactive typing.

Autocomplete-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. *Autocomplete-alternative* refers

	Fr-En	En-De	Overall
autocomplete-best	17.46	7.85	12.03
interactive typing	45.58	43.06	44.16
non-interactive typing	36.94	49.06	43.79
source suggestion	0.01	0.01	0.01
autocomplete-alternative	0.01	0.01	0.01

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

	Fr-En	En-De	Overall
autocomplete-best	71.09	60.46	65.61
interactive typing	15.92	18.37	17.18
non-interactive typing	12.90	20.93	17.04
source suggestion	0.04	0.06	0.05
autocomplete-alternative	0.05	0.19	0.12

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

to users selecting (using the mouse or down arrow) and accepting a translation from the drop-down menu. *Interactive typing* refers to users typing and modifying the last word in the partial translation, triggering real-time updates to the machine translation. *Non-interactive typing* refers to users modifying any other word in the partial translation.

We recorded over 1.1 million UI events across all translation sessions. Focusing on only PTM sessions, we identified a subset of 258,000 editing events corresponding to the five modes. We exclude non-editing events, such as hovering over the source text for word lookup without insertion. We thus measure the direct means by which the translators entered their translations. A notable shortcoming of previous systems is that users tended to eschew interactive aids in favor often typing.

Table 5 shows the proportions of editing events. Table 6 shows the total amount of text modified, measured by the number of characters entered or deleted by the users. We find that nearly two thirds (65.61%) of the text generated came from autocomplete-best at an average of 14.01 characters per keystroke. Over 88% of the editing events came from typing, but such actions accounted for only 34.22% of the text generated. While a direct comparison with previous systems is not possible, we point out the following contrasts. In the TransType system, the authors commented that their users often "[accepted] predictions in [their] entirety and then edited to ensure its correctness" and reported that 52% of target characters were typed [36]. In the "prediction+options" experiment conducted by Koehn et al. [29], the authors reported that 36% of the final translations were typed, 36% entered via a mouse click, and 27% entered via the tab key to accept machine translations. When working in our PTM system, users directly utilized machine translations to a greater degree than previously reported.

As many professional translators are touch typists, one of our design goals was retain user focus at the point of text entry and optimize text entry via the keyboard. Tables 5 and 6 show success: 99.98% of the editing events (corresponding to 99.83% of the text entered) were performed using the keyboard via autocomplete-best, interactive, or non-interactive typing.

Qualitative Analysis

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 augur 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.

REFERENCES

1. Alabau, V., Bonk, R., Buck, C., Carl, M., Casacuberta, F., García-Martínez, M., et al. Advanced computer aided translation with a web-based workbench. In *2nd Workshop on Post-Editing Technologies and Practice (WPTP)* (2013).
2. Alabau, V., González-Rubio, J., Leiva, L., Ortiz-Martínez, D., Sanchis-Trilles, G., Casacuberta, F., et al. User evaluation of advanced interaction features for a computer-assisted translation workbench. In *MT Summit XIV* (2013).
3. Alabau, V., Leiva, L. A., Ortiz-Martínez, D., and Casacuberta, F. User evaluation of interactive machine translation systems. In *EAMT* (2012).
4. Bar-Hillel, Y. The present status of automatic translation of languages. *Advances in Computers 1* (1960), 91–163.
5. Barr, D. J., Levy, R., Scheepers, C., and Tily, H. J. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language 68*, 3 (2013), 255–278.
6. Barrachina, S., Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., et al. Statistical approaches to computer-assisted translation. *Computational Linguistics 35*, 1 (2009), 3–28.
7. Bates, D. M. `lme4`: Linear mixed-effects models using S4 classes. Tech. rep., R package version 1.1-5, **`http://cran.r-project.org/package=lme4`**, 2007.
8. Bisbey, R., and Kay, M. The MIND translation system: a study in man-machine collaboration. Tech. Rep. P-4786, Rand Corp., March 1972.
9. Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., et al. Findings of the 2013 Workshop on Statistical Machine Translation. In *WMT* (2013).
10. Callison-Burch, C., Osborne, M., and Koehn, P. Re-evaluation the role of BLEU in machine translation research. In *EACL* (2006).
11. Carbonell, J. AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems 11*, 4 (1970), 190–202.
12. Carl, M. A computational framework for a cognitive model of human translation processes. In *Aslib Translating and the Computer Conference* (2010).
13. Cer, D., Manning, C. D., and Jurafsky, D. The best lexical metric for phrase-based statistical MT system optimization. In *NAACL* (2010).
14. Church, K. W., and Hovy, E. Good applications for crummy machine translation. *Machine Translation 8* (1993), 239–258.
15. Culy, C., and Riehemann, S. Z. The limits of n-gram translation evaluation metrics. In *MT Summit IX* (2003).
16. Esteban, J., Lorenzo, J., Valderrábanos, A. S., and Lapalme, G. TransType2: An innovative computer-assisted translation system. In *ACL 2004, Demonstration Session* (2004).
17. Foster, G., Isabelle, P., and Plamondon, P. Target-text mediated interactive machine translation. *Machine Translation 12*, 1/2 (1997), 175–194.
18. Foster, G., Langlais, P., and Lapalme, G. TransType: text prediction for translators. In *HLT* (2002).
19. Foster, G., Langlais, P., and Lapalme, G. User-friendly text prediction for translators. In *EMNLP* (2002).
20. Green, S., Cer, D., and Manning, C. D. Phrasal: A toolkit for new directions in statistical machine translation. In *WMT* (2014).
21. Green, S., Heer, J., and Manning, C. D. The efficacy of human post-editing for language translation. In *CHI* (2013).
22. Green, S., Wang, S., Cer, D., and Manning, C. D. Fast and adaptive online training of feature-rich translation models. In *ACL* (2013).
23. Heafield, K., Pouzyrevsky, I., Clark, J. H., and Koehn, P. Scalable modified Kneser-Ney language model estimation. In *ACL, Short Papers* (2013).
24. Horvitz, E. Principles of mixed-initiative user interfaces. In *CHI* (1999).
25. Hu, C., Bederson, B., and Resnik, P. Translation by iterative collaboration between monolingual users. In *Graphics Interface (GI)* (2010).
26. Hu, C., Resnik, P., Kronrod, Y., and Bederson, B. Deploying MonoTrans widgets in the wild. In *CHI* (2012).
27. Hwa, R., Resnik, P., Weinberg, A., and Kolak, O. Evaluating translational correspondence using annotation projection. In *ACL* (2002).
28. Kay, M. The proper place of men and machines in language translation. Tech. Rep. CSL-80-11, Xerox Palo Alto Research Center (PARC), 1980.
29. Koehn, P. A process study of computer-aided translation. *Machine Translation 23* (2009), 241–263.
30. Koehn, P. A web-based interactive computer aided translation tool. In *ACL-IJCNLP, Software Demonstrations* (2009).
31. Koehn, P. *Statistical Machine Translation*. Cambridge University Press, 2010.
32. Langlais, P., and Lapalme, G. TransType: Development-evaluation cycles to boost translator's productivity. *Machine Translation 17*, 2 (2002), 77–98.
33. Liang, P., Taskar, B., and Klein, D. Alignment by agreement. In *NAACL* (2006).
34. Lin, C.-Y., and Och, F. J. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *COLING* (2004).
35. Loh, S.-C., and Kong, L. An interactive online machine translation system (Chinese into English). In *Translating and the computer : proceedings of a seminar, London, 14th November, 1978*. North Holland, 1979.
36. Macklovitch, E. The contribution of end-users to the TransType2 project. In *Machine Translation: From Real Users to Research*, vol. 3265 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004, 197–207.
37. Macklovitch, E. TransType2: The last word. In *LREC* (2006).
38. Macklovitch, E., Nguyen, N. T., and La, G. Tracing translations in the making. In *MT Summit X* (2005).
39. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. The Stanford CoreNLP natural language processing toolkit. In *ACL, System Demonstrations* (2014).
40. Morita, D., and Ishida, T. Collaborative translation by monolinguals with machine translators. In *IUI* (2009).
41. Morita, D., and Ishida, T. Designing protocols for collaborative translation. In *Principles of Practice in Multi-Agent Systems* (2009).
42. Papineni, K., Roukos, S., Ward, T., and Zhu, W. BLEU: a method for automatic evaluation of machine translation. In *ACL* (2002).
43. Ray, R. Ten essential research findings for 2013. In *2013 Resource Directory & Index*. Multilingual, 2013.
44. Slocum, J. A survey of machine translation: Its history, current status, and future prospects. *Computational Linguistics 11*, 1 (1985), 1–17.
45. Tinkel, K. Taking it in: What makes type easy to read and why. Tech. rep., Adobe, 1996.
46. Whitelock, P. J., Wood, M. M., Ch, B. J., Holden, N., and Horsfall, H. J. Strategies for interactive machine translation: the experience and implications of the UMIST Japanese project. In *COLING* (1986), 329–334.