Human Effort and Machine Learnability in Computer Aided Translation

Spence Green, Sida Wang, Jason Chuang,[*] Jeffrey Heer,[*] Sebastian Schuster,
and **Christopher D. Manning**

Computer Science Department, Stanford University
{spenceg, sidaw, sebschu, manning}@stanford.edu

[*]Computer Science Department, University of Washington
{jcchuang, jheer}@uw.edu

Abstract

Analyses of computer aided translation typically focus on either frontend interfaces and human effort, or backend translation and machine learnability of corrections. However, this distinction is artificial in practice since the frontend and backend must work in concert. We present the first holistic, quantitative evaluation of these issues by contrasting two assistive modes: post-editing and interactive machine translation (MT). We describe a new translator interface, extensive modifications to a phrase-based MT system, and a novel objective function for re-tuning to human corrections. Evaluation with professional bilingual translators shows that post-edit is faster than interactive at the cost of translation quality for French-English and English-German. However, re-tuning the MT system to interactive output leads to larger, statistically significant reductions in HTER versus re-tuning to post-edit. Analysis shows that tuning directly to HTER results in fine-grained corrections to subsequent machine output.

1 Introduction

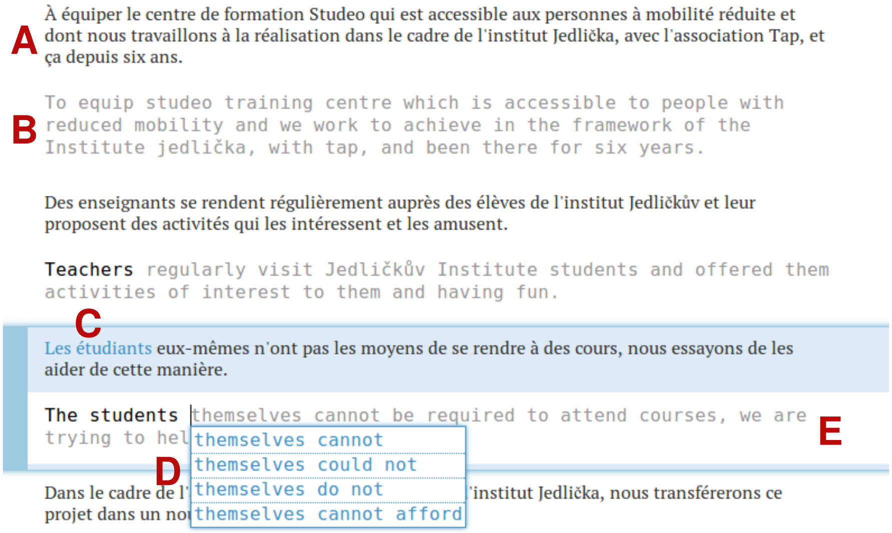
The goal of machine translation has always been to reduce human effort, whether by partial assistance or by outright replacement. However, preoccupation with the latter—fully automatic translation—at the exclusion of the former has been a feature of the research community since its first nascent steps in the 1950s. Pessimistic about progress during that decade and future prospects, Bar-Hillel (1960, p.3) argued that more attention should be paid to a "machine-post-editor partnership," whose decisive problem is "the region of optimality in the continuum of possible divisions of labor." Today, with human-quality, fully automatic machine translation (MT) elusive still, that decades-old recommendation remains current.

This paper is the first to look at both sides of the partnership in a single user study. We compare two common flavors of machine-assisted translation: post-editing and interactive MT. We analyze professional, bilingual translators working in both modes, looking first at user productivity. Does the additional machine assistance available in the interactive mode affect translation time and/or quality?

Then we turn to the machine side of the partnership. The user study results in corrections to the baseline MT output. Do these corrections help the MT system, and can it learn from them quickly enough to help the user? We perform a re-tuning experiment in which we directly optimize human Translation Edit Rate (HTER), which correlates highly with human judgments of fluency and adequacy (Snover et al., 2006). It is also an intuitive measure of human effort, making fine distinctions between 0 (no editing) and 1 (complete rewrite).

We designed a new user interface (UI) for the experiment. The interface places demands on the MT backend—not the other way around. The most significant new MT system features are *prefix decoding*, for translation completion based on a user prefix; and *dynamic phrase table augmentation*, to handle target out-of-vocabulary (OOV) words. Discriminative re-tuning is accomplished with a novel cross-entropy objective function.

We report three main findings: (1) post-editing is faster than interactive MT, corroborating Koehn (2009a); (2) interactive MT yields higher quality translation when baseline MT quality is high; and (3) re-tuning to interactive feedback leads to larger held-out HTER gains relative to post-edit. Together these results show that a human-centered approach to computer aided translation (CAT) may involve tradeoffs between human effort and machine learnability. For example, if speed is the top priority, then a design geared toward post-editing



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.

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.

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 Jedličkův Institute students and offered them activities of interest to them and having fun.

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 themselves cannot be required to attend courses, we are trying to hel

Dans le cadre de l' 'institut Jedlička, nous transférerons ce projet dans un no

Figure 1: Main translation interface. The user sees the full document context, with French source inputs (A) interleaved with suggested English translations (B). The sentence in focus is indicated by the blue rectangle, which can be moved via two hot keys. Source coverage (C) of the user prefix—shaded in blue—updates as the user works, as do autocomplete suggestions (D) and a full completion (E).

is appropriate. However, if reductions in HTER ultimately correspond to lower human effort, then investing slightly more time in the interactive mode, which results in more learnable output, may be optimal. Mixed UI designs may offer a compromise. Code and data from our experiments are available at: `http://nlp.stanford.edu/software/phrasal/`

A holistic comparison with human subjects necessarily involves many moving parts. Section 2 briefly describes the interface, focusing on NLP components. Section 3 describes changes to the backend MT system. Section 4 explains the user study, and reports human translation time and quality results. Section 5 describes the MT re-tuning experiment. Analysis (section 6) and related work (section 7) round out the paper.

2 New Translator User Interface

Figure 1 shows the translator interface, which is designed for expert, bilingual translators. Previous studies have shown that expert translators work and type quickly (Carl, 2010), so the interface is designed to be very responsive, and to be primarily operated by the keyboard. Most aids can be accessed via either typing or four 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 tools.

In the trivial post-edit mode, the interactive aids are disabled and a 1-best translation pre-populates the text entry box.

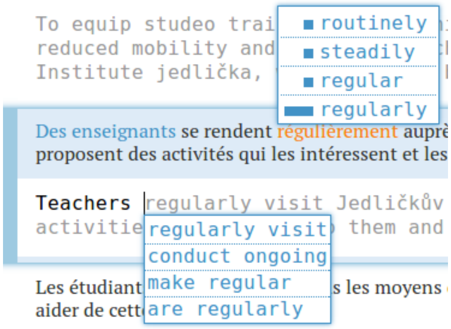
We have described the HCI-specific motivations for and contributions of this new interface in Green et al. (2014c). This section focuses on interface elements built on NLP components.

2.1 UI Overview and Walkthrough

We categorized interactions into three groups: **source comprehension**: word lookups, *source coverage highlighting*; **target gisting**: 1-best translation, *real-time target completion*; **target generation**: real-time autocomplete, *target reordering*, insert complete translation. The interaction designs are novel; those in *italic* have, to our knowledge, never appeared in a translation workbench.

Source word lookup When the user hovers over a source word, a menu of up to four ranked translation suggestions appears (Figure 2). The menu is populated by a phrase-table query of the word plus one token of left context. This query usually returns in under 50ms. The width of the horizontal bars indicates confidence, with the most confident suggestion 'regularly' placed at the bottom, nearest to the cursor. The user can insert a translation suggestion by clicking.

Source coverage highlighting The source coverage feature (Figure 1C) helps the user quickly find untranslated words in the source. The interaction is



To equip studeo trai
reduced mobility and
Institute jedlička,

Des enseignants se rendent régulièrement auprè
proposent des activités qui les intéressent et les

Teachers regularly visit Jedličkův
activitie them and

Les étudiant s les moyens
aider de cett

Figure 2: Source word lookup and target autocomplete menus. The menus show different suggestions. The word lookup menu (top) is not dependent on the target context *Teachers*, whereas the autocomplete dropdown (bottom) is.

based on the word alignments between source and target generated by the MT system. We found that the raw alignments are too noisy to show users, so the UI filters them with phrase-level heuristics.

1-best translation The most common use of MT output is *gisting* (Koehn, 2010, p.21). The gray text below each black source input shows the best MT system output (Figure 1B).

Real-time target completion When the user extends the black prefix, the gray text will update to the most probable completion (Figure 1E). This update comes from decoding under the full translation model. All previous systems performed inference in a word lattice.

Real-time autocomplete The autocomplete dropdown at the point of text entry is the main translation aid (Figures 1D and 2). Each real-time update actually contains a distinct 10-best list for the full source input. The UI builds up a trie from these 10-best lists. Up to four distinct suggestions are then shown at the point of translation. The suggestion length is based on a syntactic parse of the fixed source input. As an offline, pre-processing step, we parse each source input with Stanford CoreNLP (Manning et al., 2014). The UI combines those parses with word alignments from the full translation suggestions to project syntactic constituents to each item on the n-best list. Syntactic projection is a very old idea that underlies many MT systems (see: Hwa et al. (2002)). Here we make novel use of it for suggestion prediction filtering.[1] Presently, we project noun phrases, verb phrases (minus the verbal arguments), and prepositional phrases. Crucially, these units are natural to humans, unlike statistical target phrases.

Target Reordering Carl (2010) showed that expert translators tend to adopt *local planning*: they read a few words ahead and then translate in a roughly online fashion. However, word order differences between languages will necessarily require longer range planning and movement. To that end, the UI supports keyboard-based reordering. Suppose that the user wants to move a span in gray text to the insertion position for editing. Typing the prefix of this string 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.

Insert complete translation The user can insert the full completion via a hot key. Notice that if the user presses this hot key immediately, all gray text becomes black, and the interface effectively switches to post-edit mode. This feature greatly accelerates translation when the MT is mostly correct, and the user only wants to make a few changes.

2.2 User Activity Logging

A web application serves the Javascript-based interface, relays translation requests to the MT system, and logs user records to a database. Each user record is a tuple of the form $(f, \hat{e}, h, u)$, where f is the source sequence, $\hat{e}$ is the latest 1-best machine translation of f, h is the correction of $\hat{e}$, and u is the log of interaction events during the translation session. Our evaluation corpora also include independently generated references e for each f.

3 Interactive MT Backend

Now we describe modifications to Phrasal (Green et al., 2014b), the phrase-based MT system that supports the interface. Phrasal follows the log-linear approach to phrase-based translation (Och and Ney, 2004) in which the decision rule has the familiar linear form

$$\hat{e} = \arg\max_{e} w^{\top} \phi(e, f) \qquad (1)$$

[1] The classic TransType system included a probabilistic prediction length component (Foster et al., 2002), but we find that the simpler projection technique works well in practice.

where $w \in \mathbb{R}^d$ is the model weight vector and $\phi(\cdot) \in \mathbb{R}^d$ is a feature map.

3.1 Decoding

The default Phrasal search algorithm is cube pruning (Huang and Chiang, 2007). In the post-edit condition, search is executed as usual for each source input, and the 1-best output is inserted into the target textbox. However, in interactive mode, the full search algorithm is executed *each time* the user modifies the partial translation. Machine suggestions $\hat{e}$ must match user prefix h. Define indicator function $\text{pref}(\hat{e}, h)$ to return true if $\hat{e}$ begins with h, and false otherwise. Eq. 1 becomes:

$$\hat{e} = \underset{e \text{ s.t. } \text{pref}(e,h)}{\arg\max} \; w^\top \phi(e, f) \qquad (2)$$

Cube pruning can be straightforwardly modified to satisfy this constraint by simple string matching of candidate translations. Also, the pop limit must be suspended until at least one legal candidate appears on each beam, or the priority queue of candidates is exhausted. We call this technique *prefix decoding*.[2]

There is another problem. Human translators are likely to insert unknown target words, including new vocabulary, misspellings, and typographical errors. They might also reorder source text so as to violate the phrase-based distortion limit. To solve these problems, we perform *dynamic phrase table augmentation*, adding new synthetic rules specific to each search. Rules allowing any source word to align with any unseen or ungeneratable (due to the distortion limit) target word are created.[3] These synthetic rules are given rule scores lower than any other rules in the set of queried rules for that source input f. Then candidates are allowed to compete on the beam. Candidates with spurious alignments will likely be pruned in favor of those that only turn to synthetic rules as a last resort.

3.2 Tuning

We choose BLEU (Papineni et al., 2002) for baseline tuning to independent references, and HTER for re-tuning to human corrections. Our rationale is as follows: Cer et al. (2010) showed that BLEU-tuned systems score well across automatic metrics and also correlate with human judgment better than systems tuned to other metrics. Conversely, systems tuned to edit-distance-based metrics like TER tend to produce short translations that are heavily penalized by other metrics.

When human corrections become available, we switch to HTER, which correlates with human judgment and is an interpretable measure of editing effort. Whereas TER is computed as $\text{TER}(e, \hat{e})$, HTER is $\text{HTER}(h, \hat{e})$. HBLEU is an alternative, but since BLEU is invariant to some permutations (Callison-Burch et al., 2006), it is less interpretable. We find that it also does not work as well in practice.

We previously proposed a fast, online tuning algorithm (Green et al., 2013b) based on AdaGrad (Duchi et al., 2011). The default loss function is expected error (EE) (Och, 2003; Cherry and Foster, 2012). Expected BLEU is an example of EE, which we found to be unstable when switching metrics. This may result from direct incorporation of the error metric into the gradient computation.

To solve this problem, we propose a *cross-entropy loss* which, to our knowledge, is new in MT. Let $\hat{E} = \{\hat{e}_i\}_{i=1}^n$ be an n-best list ranked by a gold metric $G(e, \hat{e}) \geq 0$. Assume we have a preference of a higher G (e.g., BLEU or $1-$HTER). Define the model distribution over $\hat{E}$ as $q(\hat{e}|f) \propto \exp[w^\top \phi(\hat{e}, f)]$ normalized so that $\sum_{\hat{e} \in \hat{E}} q(\hat{e}|f) = 1$; q indicates how much the model prefers each translation. Similarly, define $p(\hat{e}|f)$ based on any function of the gold metric so that $\sum_{\hat{e} \in \hat{E}} p(\hat{e}|f) = 1$; p indicates how much the metric prefers each translation. We choose a DCG-style[4] parameterization that skews the p distribution toward higher-ranked items on the n-best list: $p(\hat{e}_i|f) \propto G(e, \hat{e}_i)/\log(1 + i)$ for the ith ranked item. The cross-entropy (CE) loss function is:

$$\ell_{\text{CE}}(w; \hat{E}) = \mathbb{E}_{p(\hat{e}|f)}[-\log(q(\hat{e}|f)] \qquad (3)$$

It turns out that if p is simply the posterior distribution of the metric, then this loss is related to the log of the standard EE loss:[5]

$$\ell_{\text{EE}}(w; \hat{E}) = -\log[\mathbb{E}_{p(\hat{e}|f)}[q(\hat{e}|f)]] \qquad (4)$$

We can show that $\ell_{\text{CE}} \geq \ell_{\text{EE}}$ by applying Jensen's inequality to the function $-\log(\cdot)$. So minimizing ℓ_{CE} also minimizes a convex upper bound of the log expected error. This convexity given the n-

[2] Och et al. (2003) describe a similar algorithm for word graphs.

[3] Ortiz-Martínez et al. (2009) describe a related technique in which *all* source and target words can align, with scores set by smoothing.

[4] Discounted cumulative gain (DCG) is widely used in information retrieval learning-to-rank settings. n-best MT learning is standardly formulated as a ranking task.

[5] For expected error, $p(\hat{e}_i) = G(e, \hat{e}_i)$ is not usually normalized. Normalizing p adds a negligible constant.

best list does not mean that the overall MT tuning loss is convex, since the n-best list contents and order depend on the parameters w. However, all regret bounds and other guarantees of online convex optimization would now apply in the CE case since $\ell_{\text{CE},t}(w_{t-1}; E_t)$ is convex for each t. This is attractive compared to expected error, which is non-convex even given the n-best list. We empirically observed that CE converges faster and is less sensitive to hyperparameters than EE.

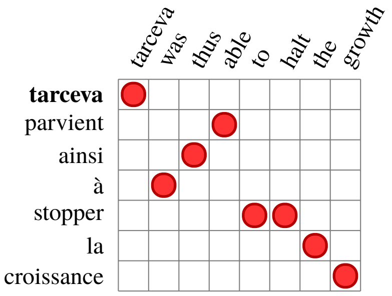
Faster decoding trick We found that online tuning also permits a trick that speeds up decoding during deployment. Whereas the Phrasal default beam size is 1,200, we were able to reduce the beam size to 800 and run the tuner longer to achieve the same level of translation quality. For example, at the default beam size for French-English, the algorithm converges after 12 iterations, whereas at the lower beam size it achieves that level after 20 iterations. In our experience, batch tuning algorithms seem to be more sensitive to the beam size.

3.3 Feature Templates

The baseline system contains 19 dense feature templates: the nine Moses (Koehn et al., 2007) baseline features, the eight-feature hierarchical lexicalized re-ordering model of Galley and Manning (2008), the (log) count of each rule in the bitext, and an indicator for unique rules. We found that sparse features, while improving translation quality, came at the cost of slower decoding due to feature extraction and inner products with a higher dimensional feature map ϕ. During prototyping, we observed that users found the system to be sluggish unless it responded in approximately 300ms or less. This budget restricted us to dense features.

When re-tuning to corrections, we extract features from the user logs u and add them to the baseline dense model. For each tuning input f, the MT system produces candidate derivations $d = (f, \hat{e}, a)$, where a is a word alignment. The user log u also contains the *last MT derivation*[6] accepted by the user $d_u = (f, \hat{e}_u, a_u)$. We extract features by comparing d and d_u. The heuristic we take is *intersection*: $\phi(d) \leftarrow \phi(d) \cap \phi(d_u)$.

Lexicalized and class-based alignments Consider the alignment in Figure 3. We find that user derivations often contain many unigram rules,



	tarceva	was	thus	able	to	halt	the	growth
tarceva	●							
parvient				●				
ainsi			●					
à		●						
stopper					●	●		
la							●	
croissance								●

Figure 3: User translation word alignment obtained via prefix decoding and dynamic phrase table augmentation.

which are less powerful than larger phrases, but nonetheless provide high-precision lexical choice information. We fire indicators for both unigram links and multiword cliques. We also fire class-based versions of this feature.

Source OOV blanket Source OOVs are usually more frequent when adapting to a new domain. In the case of European languages—our experimental setting—many of the words simply transfer to the target, so the issue is where to position them. In Figure 3, the proper noun *tarceva* is unknown, so the decoder OOV model generates an identity translation rule. We add features in which the source word is concatenated with the left, right, and left/right contexts in the target, e.g., {`<s>-tarceva`, `tarceva-was`, `<s>-tarceva-was`}. We also add versions with target words mapped to classes.

3.4 Differences from Previous Work

Our backend innovations support the UI and enable feature-based learning from human corrections. In contrast, most previous work on incremental MT learning has focused on extracting new translation rules, language model updating, and modifying translation model probabilities (see: Denkowski et al. (2014a)). We regard these features as additive to our own work: certainly extracting new, unseen rules should help translation in a new domain. Moreover, to our knowledge, all previous work on updating the weight vector w has considered *simulated post-editing*, in which the independent references e are substituted for corrections h. Here we extract features from and re-tune to actual corrections to the baseline MT output.

[6] Extracting features from intermediate user editing actions is an interesting direction for future work.

4 Translation User Study

We conducted a human translation experiment with a 2 (translation conditions) × n (source sentences) mixed design, where n depended on the language pair. Translation conditions (post-edit and 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 influence another.

Subjects completed the experiment remotely on their own hardware. They received personalized login credentials for the translation interface, which administered the experiment. Subjects first completed a demographic questionnaire about prior experience with CAT and language proficiency. Next, they completed a training module that included a 4-minute tutorial video and a practice “sandbox” for developing proficiency with the UI. Then subjects completed the translation experiment. Finally, they completed an exit questionnaire.

Unlike the experiment of Koehn (2009a), subjects were under time pressure. An idle timer prevented subjects from pausing for more than three minutes while the translator interface was open. This constraint eliminates a source of confound in the timing analysis.

We randomized the order of translation conditions and the assignment of sentences to conditions. At most five sentences appeared per screen, and those sentences appeared in the source document order. Subjects could move among sentences within a screen, but could not revise previous screens. Subjects received untimed breaks both between translation conditions and after about every five screens within a translation condition.

4.1 Linguistic Materials

We chose two language pairs: French-English (Fr-En) and English-German (En-De). Anecdotally, French-English is an easy language pair for MT, whereas English-German is very hard due to reordering and complex German morphology.

We chose three text genres: software, medical, and informal news. The software text came from the graphical interfaces of Autodesk AutoCAD and Adobe Photoshop. The medical text was a drug review from the European Medicines Agency. These two data sets came from TAUS[7] and included independent reference translations. The informal news text came from the WMT 2013 shared task test set (Bojar et al., 2013). The evaluation corpus was constructed from equal proportions of the three genres.

The Fr-En dataset contained 3,003 source tokens (150 segments); the En-De dataset contained 3,002 (173 segments). As a rule of thumb, a human translator averages about 2,700 source tokens per day (Ray, 2013, p.36), so the experiment was designed to replicate a slightly demanding work day.

4.2 Selection of Subjects

For each language pair, we recruited 16 professional, freelance translators on Proz, which is the largest online translation community.[8] We posted ads for both language pairs at a fixed rate of $0.085 per source word, an average rate in the industry. In addition, we paid $10 to each translator for completing the training module. All subjects had significant prior experience with a CAT workbench.

4.3 Results

We analyze the translation conditions in terms of two response variables: time and quality. 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 had a technical problem that prevented logging. Finally, we also filtered segment-level sessions for which the log of translation time was greater than 2.5 standard deviations from the mean.

4.3.1 Translation Time

We analyze time with a linear mixed effects model (LMEM) estimated with the `lme4` (Bates, 2007) R package. When experimental factors are sampled from larger populations—e.g., humans, sentences, words—LMEMs are more robust to type II errors (see: Baayen et al. (2008)). The log-transformed time is the response variable and translation condition is the main independent variable. The maximal random effects structure (Barr et al., 2013) contains intercepts for subject, sentence id, and text genre, each with random slopes for translation condition.

We found significant main effects for translation condition (Fr-En, $p < 0.05$; En-De, $p < 0.01$). The orientation of the coefficients indicates that interactive is 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

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

[8] `http://www.proz.com`

	Fr-En		En-De	
	TER	**HTER**	**TER**	**HTER**
post-edit	47.32	23.51	56.16	37.15
interactive	47.05	24.14	55.89	39.55

Table 1: Automatic assessment of translation quality. Here we change the definitions of TER and HTER slightly. TER is the human translations compared to the independent references. HTER is the baseline MT compared to the human corrections.

in interactive, or 18.7% slower. For En-De, the mean is 51.8 sec/sentence vs. 63.3 sec/sentence in interactive, or 22.1% slower.

We found other predictive covariates that reveal more about translator behavior. When subjects did not edit the MT suggestion, they were significantly faster. When token edit distance from MT or source input length increased, they were slower. Subjects were usually faster as the experiment progressed, a result that may indicate increased proficiency with practice. Note that all subjects reported professional familiarity with post-edit, whereas the interactive mode was entirely new to them. In the exit survey many translators suggested that with more practice, they could have been as fast in the interactive mode.[9]

4.3.2 Translation Quality

We evaluated translation quality with both automatic and manual measures. Table 1 shows that in the interactive mode, TER is lower and HTER is higher: subjects created translations closer to the references (lower TER), but performed more editing (higher HTER). This result suggests better translations in the interactive mode.

To confirm that intuition, we elicited judgments from professional human raters. The setup followed the manual quality evaluation of the WMT 2014 shared task (Bojar et al., 2014). We hired six raters—three for each language pair—who were paid between $15–20 per hour. The raters logged into Appraise (Federmann, 2010) and for each source segment, ranked five randomly selected translations. From these 5-way rankings we extracted pairwise judgments $\pi = \{<, =\}$, where $u_1 < u_2$ indicates that subject u_1 provided a better translation than subject u_2 for a given source input (Table 2).

	Fr-En		**En-De**	
#pairwise	14,211		15,001	
#ties (=)	5,528		2,964	
IAA	0.419	(0.357)	0.407	(0.427)
EW (inter.)	0.512		0.491	

Table 2: Pairwise judgments for the manual quality assessment. Inter-annotator agreement (*IAA*) κ scores are measured with the official WMT14 script. For comparison, the WMT14 IAA scores are given in parentheses. *EW (inter.)* is expected wins of interactive according to Eq. (6).

	Fr-En		**En-De**	
	sign	p	sign	p
ui (interactive)	+	•	−	
log edit distance	−	•••	+	•••
gender (female)	−		+	•
log session order	−		+	•

Table 3: LMEM manual translation quality results for each fixed effect with contrast conditions for binary predictors in (). The signs of the coefficients can be interpreted as in ordinary regression. *edit distance* is token-level edit distance from baseline MT. *session order* is the order in which the subject translated the sentence during the experiment. Statistical significance was computed with a likelihood ratio test: ••• $p < 0.001$; • $p < 0.05$.

In WMT the objective is to rank individual systems; here we need only compare interface conditions. However, we should control for translator variability. Therefore, we build a binomial LMEM for quality. The model is motivated by the simple and intuitive *expected wins* (EW) measure used at WMT. Let S be the set of pairwise judgments and $\text{wins}(u_1, u_2) = |\{(u_1, u_2, \pi) \in S \mid \pi = <\}|$. The standard EW measure is:

$$e(u_1) = \frac{1}{|S|} \sum_{u_1 \neq u_2} \frac{\text{wins}(u_1, u_2)}{\text{wins}(u_1, u_2) + \text{wins}(u_2, u_1)} \tag{5}$$

Sakaguchi et al. (2014) showed that, despite its simplicity, Eq. (5) is nearly as effective as model-based methods given sufficient high-quality judgments. Since we care only about the two translation conditions, we reinterpret the u_i as interface conditions, i.e., $u_1 = \text{int}$ and $u_2 = \text{pe}$. We can then disregard

[9] See (Green et al., 2014c) for significance levels of the other covariates along with analysis of subject learning rates, subject behavior, and qualitative feedback.

the normalizing term to obtain:

$$e(u_1) = \frac{\text{wins}(u_1, u_2)}{\text{wins}(u_1, u_2) + \text{wins}(u_2, u_1)} \qquad (6)$$

which is the expected value of a Bernoulli distribution (so $e(u_2) = 1 - e(u_1)$). The intercept-term of the binomial LMEM will be approximately this value subject to other fixed and random effects.

To estimate the model, we convert each pairwise judgment $u_1 < u_2$ to two examples where the response is 1 for u_1 and 0 for u_2. We add the fixed effects shown in Table 3, where the numeric effects are centered and scaled by their standard deviations. The maximal random effects structure contains intercepts for sentence id nested within subject along with random slopes for interface condition.

Table 3 shows the p-values and coefficient orientations. The models yield probabilities that can be interpreted like Eq. (6) but with all fixed predictors set to 0. For Fr-En, the value for post-edit is 0.472 vs. 0.527 for interactive. For En-De, post-edit is 0.474 vs. 0.467 for interactive. The difference is statistically significant for Fr-En, but not for En-De.

When MT quality was anecdotally high (Fr-En), high token-level edit distance from the initial suggestion decreased quality. When MT was poor (En-De), significant editing improved quality. Female En-De translators were better than males, possibly due to imbalance in the subject pool (12 females vs. 4 males). En-De translators seemed to improve with practice (positive coefficient for *session order*).

The Fr-En results are the first showing an interactive UI that improves over post-edit.

5 MT Re-tuning Experiment

The human translators corrected the output of the BLEU-tuned, baseline MT system. No updating of the MT system occurred during the experiment to eliminate a confound in the time and quality analyses. Now we investigate re-tuning the MT system to the corrections by simply re-starting the online learning algorithm from the baseline weight vector w, this time scoring with HTER instead of BLEU.

Conventional incremental MT learning experiments typically resemble domain adaptation: small-scale baselines are trained and tuned on mostly out-of-domain data, and then re-tuned incrementally on in-domain data. In contrast, we start with large-scale systems. This is more consistent with a professional translation environment where translators receive suggestions from state-of-the-art systems like Google Translate.

	Bilingual		**Monolingual**
	#Segments	***#Tokens***	***#Tokens***
En-De	4.54M	224M	1.7B
Fr-En	14.8M	842M	2.24B

Table 4: Gross statistics of MT training corpora.

	En-De	**Fr-En**
baseline-tune	9,469	8,931
baseline-dev	9,012	9,030
int-tune	680	589
int-test	457	368
pe-tune	764	709
pe-test	492	447

Table 5: Tuning, development, and test corpora (#segments). **tune** and **dev** were used for baseline system preparation. Re-tuning was performed on **int-tune** and **pe-tune**, respectively. We report held-out results on the two test data sets. All sets are supplied with independent references.

5.1 Datasets

Table 4 shows the monolingual and parallel training corpora. Most of the data come from the constrained track of the WMT 2013 shared task (Bojar et al., 2013). 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 (Liang et al., 2006) and symmetrized the alignments with the grow-diag heuristic. For each target language we used lmplz (Heafield et al., 2013) to estimate unfiltered, 5-gram Kneser-Ney LMs from the concatenation of the target side of the bitext and the monolingual data. For the class-based features, we estimated 512-class source and target mappings with the algorithm of Green et al. (2014a).

The upper part of Table 5 shows the baseline tuning and development sets, which also contained 1/3 TAUS medical text, 1/3 TAUS software text, and 1/3 WMT newswire text (see section 4).

The lower part of Table 5 shows the organization of the human corrections for re-tuning and testing. Recall that for each unique source input, eight human translators produced a correction in each condition. First, we filtered all corrections for which a log u was not recorded (due to technical problems). Second, we de-duplicated the corrections so that each h was unique. Finally, we split the unique (f, h) tuples according to a natural division in the

System	tune	BLEU↑	TER↓	HTER
baseline	bleu	23.12	60.29	44.05
re-tune	hter	22.18	60.85	43.99
re-tune+feat	hter	21.73	*59.71*	**42.35**

(a) En-De int-test results.

System	tune	BLEU↑	TER↓	HTER
baseline	bleu	39.33	45.29	28.28
re-tune	hter	39.99	45.73	26.96
re-tune+feat	hter	*40.30*	45.28	**26.40**

(b) Fr-En int-test results.

Table 6: Main re-tuning results for interactive data. **baseline** is the BLEU-tuned system used in the translation user study. **re-tune** is the baseline feature set re-tuned to HTER on int-tune. **re-tune+feat** adds the human feature templates described in section 3.3. **bold** indicates statistical significance relative to the baseline at $p < 0.001$; *italic* at $p < 0.05$ by the permutation test of Riezler and Maxwell (2005).

System	HTER↓ int	System	HTER↓ pe
baseline	44.05	baseline	41.05
re-tune (int)	43.99	re-tune (pe)	**40.34**
re-tune+feat	**42.35**	–	–
Δ	-1.80		-0.71

Table 7: En-De test results for re-tuning to post-edit (pe) vs. interactive (int). Features cannot be extracted from the post-edit data, so the re-tune+feat system cannot be learned. The Fr-En results are similar but are omitted due to space.

data. There were five source segments per document, and each document was rendered as a single screen during the translation experiment. *Segment order was not randomized*, so we could split the data as follows: assign the first three segments of each screen to tune, and the last two to test. This is a clean split with no overlap.

This tune/test split has two attractive properties. First, if we can quickly re-tune on the first few sentences on a screen and provide better translations for the last few, then presumably the user experience improves. Second, source inputs f are repeated—eight translators translated each input in each condition. This means that a reduction in HTER means better average suggestions for multiple human translators. Contrast this experimental design with tuning to the corrections of a single human translator. There the system might overfit to one human style, and may not generalize to other human translators.

5.2 Results

Table 6 contains the main results for re-tuning to interactive MT corrections. For both language pairs, we observe large statistically significant reductions in HTER. However, the results for BLEU and TER—which are computed with respect to the independent references—are mixed. The lower En-De BLEU score is explained by a higher brevity penalty for the re-tuned output (0.918 vs. 0.862). However, the re-tuned 4-gram and 3-gram precisions are significantly higher. The unchanged Fr-En TER value can be explained by the observation that no human translators produced TER scores higher than the baseline MT. This odd result has also been observed for BLEU (Culy and Riehemann, 2003), although here we do observe a slight BLEU improvement.

The additional features (854 for Fr-En; 847 for En-De) help significantly and do not slow down decoding. We used the same L_1 regularization strength as the baseline, but feature growth could be further constrained by increasing this parameter. Tuning is very fast at about six minutes *for the whole dataset*, so tuning during a live user session is already practical.

Table 7 compares re-tuning to interactive vs. post-edit corrections. Recall that the int-test and pe-test datasets are different and contain different references. The post-edit baseline is lower because humans performed less editing in the baseline condition (see Table 1). Features account for the greatest reduction in HTER. Of course, the features are based mostly on word alignments, which could be obtained for the post-edit data by running an online word alignment tool (see: Farajian et al. (2014)). However, the interactive logs contain much richer user state information that we could not exploit due to data sparsity. We also hypothesize that the final interactive corrections might be more useful since suggestions prime translators (Green et al., 2013a), and the MT system was able to refine its suggestions.

6 Re-tuning Analysis

Tables 6 and 7 raise two natural questions: what accounts for the reduction in HTER, and why are the TER/BLEU results mixed? Comparison of the BLEU-tuned baseline to the HTER re-tuned systems gives some insight. For both questions, fine-grained corrections appear to make the difference.

Consider this French test example (with gloss):

(1) une ligne de chimiothérapie antérieure
one line of chemotherapy previous

The independent reference for *une ligne de chimiothérapie* is ‘previous chemotherapy treatment’, and the baseline produces ‘previous chemotherapy line.’ The source sentence appears seven times with the following user translations: ‘one line or more of chemotherapy’, ‘one prior line of chemotherapy’, ‘one previous line of chemotherapy’ (2), ‘one line of chemotherapy before’ (2), ‘one protocol of chemotherapy’. The re-tuned, feature-based system produces ‘one line of chemotherapy before’, matching two of the humans exactly, and six of the humans in terms of idiomatic medical jargon (‘line of chemotherapy’ vs. ‘chemotherapy treatment’). However, the baseline output would have received better BLEU and TER scores.

Sometimes re-tuning improves the translations with respect to both the reference and the human corrections. This English phrase appears in the En-De test set:

(2) depending on the file
abhängig von der datei

The baseline produces exactly the gloss shown in Ex. (2). The human translators produced: ‘je nach datei’ (6), ‘das dokument’, and ‘abhängig von der datei’. The re-tuned system rendered the phrase ‘je nach dokument’, which is closer to both the independent reference ‘je nach datei’ and the human corrections. This change improves TER, BLEU, and HTER.

7 Related Work

The process study most similar to ours is that of Koehn (2009a), who compared scratch, post-edit, and simple interactive modes. However, he used undergraduate, non-professional subjects, and did not consider re-tuning. Our experimental design with professional bilingual translators follows our previous work Green et al. (2013a) comparing scratch translation to post-edit.

Many research translation UIs have been proposed including TransType (Langlais et al., 2000), Caitra (Koehn, 2009b), Thot (Ortiz-Martínez and Casacuberta, 2014), TransCenter (Denkowski et al., 2014b), and CasmaCat (Alabau et al., 2013). However, to our knowledge, none of these interfaces were explicitly designed according to mixed-initiative principles from the HCI literature.

Incremental MT learning has been investigated several times, usually starting from no data (Barrachina et al., 2009; Ortiz-Martínez et al., 2010), via simulated post-editing (Martínez-Gómez et al., 2012; Denkowski et al., 2014a), or via re-ranking (Wäschle et al., 2013). No previous experiments combined large-scale baselines, full re-tuning of the model weights, and HTER optimization.

HTER tuning can be simulated by re-parameterizing an existing metric. Snover et al. (2009) tuned TERp to correlate with HTER, while Denkowski and Lavie (2010) did the same for METEOR. Zaidan and Callison-Burch (2010) showed how to solicit MT corrections for HTER from Amazon Mechanical Turk.

Our learning approach is related to coactive learning (Shivaswamy and Joachims, 2012). Their basic preference perceptron updates toward a correction, whereas we use the correction for metric scoring and feature extraction.

8 Conclusion

We presented a new CAT interface that supports post-edit and interactive modes. Evaluation with professional, bilingual translators showed post-edit to be faster, but prior subject familiarity with post-edit may have mattered. For French-English, the interactive mode enabled higher quality translation. Re-tuning the MT system to interactive corrections also yielded large HTER gains. Technical contributions that make re-tuning possible are a cross-entropy objective, prefix decoding, and dynamic phrase table augmentation. Larger quantities of corrections should yield further gains, but our current experiments already establish the feasibility of Bar-Hillel’s virtuous “machine-post-editor partnership” which benefits both humans and machines.

Acknowledgements

We thank TAUS for access to their data repository. We also thank John DeNero, Chris Dyer, Alon Lavie, and Matt Post for helpful conversations. 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

V. Alabau, R. Bonk, C. Buck, M. Carl, F. Casacuberta, M. García-Martínez, et al. 2013. Advanced computer aided translation with a web-based workbench. In *2nd Workshop on Post-Editing Technologies and Practice*.

R.H. Baayen, D.J. Davidson, and D.M. Bates. 2008. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59(4):390–412.

Y. Bar-Hillel. 1960. The present status of automatic translation of languages. *Advances in Computers*, 1:91–163.

D. J. Barr, R. Levy, C. Scheepers, and H. J. Tily. 2013. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, 68(3):255–278.

S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, et al. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

D. M. Bates. 2007. `lme4`: Linear mixed-effects models using S4 classes. Technical report, R package version 1.1-5, `http://cran.r-project.org/package=lme4`.

O. Bojar, C. Buck, C. Callison-Burch, C. Federmann, B. Haddow, P. Koehn, et al. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *WMT*.

O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, et al. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *WMT*.

C. Callison-Burch, M. Osborne, and P. Koehn. 2006. Re-evaluating the role of BLEU in machine translation research. In *EACL*.

M. Carl. 2010. A computational framework for a cognitive model of human translation processes. In *Aslib Translating and the Computer Conference*.

D. Cer, C. D. Manning, and D. Jurafsky. 2010. The best lexical metric for phrase-based statistical MT system optimization. In *NAACL*.

C. Cherry and G. Foster. 2012. Batch tuning strategies for statistical machine translation. In *NAACL*.

C. Culy and S. Z. Riehemann. 2003. The limits of n-gram translation evaluation metrics. In *MT Summit IX*.

M. Denkowski and A. Lavie. 2010. Extending the METEOR machine translation evaluation metric to the phrase level. In *NAACL*.

M. Denkowski, C. Dyer, and A. Lavie. 2014a. Learning from post-editing: Online model adaptation for statistical machine translation. In *EACL*.

M. Denkowski, A. Lavie, I. Lacruz, and C. Dyer. 2014b. Real time adaptive machine translation for post-editing with cdec and TransCenter. In *Workshop on Humans and Computer-assisted Translation*.

J. Duchi, E. Hazan, and Y. Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159.

M. A. Farajian, N. Bertoldi, and M. Federico. 2014. Online word alignment for online adaptive machine translation. In *Workshop on Humans and Computer-assisted Translation*.

C. Federmann. 2010. Appraise: An open-source toolkit for manual phrase-based evaluation of translations. In *LREC*.

G. Foster, P. Langlais, and G. Lapalme. 2002. User-friendly text prediction for translators. In *EMNLP*.

M. Galley and C. D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *EMNLP*.

S. Green, J. Heer, and C. D. Manning. 2013a. The efficacy of human post-editing for language translation. In *CHI*.

S. Green, S. Wang, D. Cer, and C. D. Manning. 2013b. Fast and adaptive online training of feature-rich translation models. In *ACL*.

S. Green, D. Cer, and C. D. Manning. 2014a. An empirical comparison of features and tuning for phrase-based machine translation. In *WMT*.

S. Green, D. Cer, and C. D. Manning. 2014b. Phrasal: A toolkit for new directions in statistical machine translation. In *WMT*.

S. Green, J. Chuang, J. Heer, and C. D. Manning. 2014c. Predictive Translation Memory: A mixed-initiative system for human language translation. In *UIST*.

K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *ACL, Short Papers*.

L. Huang and D. Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *ACL*.

R. Hwa, P. Resnik, A. Weinberg, and O. Kolak. 2002. Evaluating translational correspondence using annotation projection. In *ACL*.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL, Demonstration Session*.

P. Koehn. 2009a. A process study of computer-aided translation. *Machine Translation*, 23:241–263.

P. Koehn. 2009b. A web-based interactive computer aided translation tool. In *ACL-IJCNLP, Software Demonstrations*.

P. Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

P. Langlais, G. Foster, and G. Lapalme. 2000. TransType: a computer-aided translation typing system. In *Workshop on Embedded Machine Translation Systems*.

P. Liang, B. Taskar, and D. Klein. 2006. Alignment by agreement. In *NAACL*.

C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL, System Demonstrations*.

P. Martínez-Gómez, G. Sanchis-Trilles, and F. Casacuberta. 2012. Online adaptation strategies for statistical machine translation in post-editing scenarios. *Pattern Recognition*, 45(9):3193–3203.

F. J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

F. J. Och, R. Zens, and H. Ney. 2003. Efficient search for interactive statistical machine translation. In *EACL*.

F. J. Och. 2003. Minimum error rate training for statistical machine translation. In *ACL*.

D. Ortiz-Martínez and F. Casacuberta. 2014. The new Thot toolkit for fully automatic and interactive statistical machine translation. In *EACL, System Demonstrations*.

D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta. 2009. Interactive machine translation based on partial statistical phrase-based alignments. In *RANLP*.

D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta. 2010. Online learning for interactive statistical machine translation. In *NAACL*.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

R. Ray. 2013. Ten essential research findings for 2013. In *2013 Resource Directory & Index*. Multilingual.

S. Riezler and J. T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing in MT. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

K. Sakaguchi, M. Post, and B. Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *WMT*.

P. Shivaswamy and T. Joachims. 2012. Online structured prediction via coactive learning. In *ICML*.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *AMTA*.

M. Snover, N. Madnani, B. Dorr, and R. Schwartz. 2009. Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric. In *WMT*.

K. Wäschle, P. Simianer, N. Bertoldi, S. Riezler, and M. Federico. 2013. Generative and discriminative methods for online adaptation in SMT. In *MT Summit XIV*.

O. F. Zaidan and C. Callison-Burch. 2010. Predicting human-targeted translation edit rate via untrained human annotators. In *NAACL*.