Beyond Prefix-Based Interactive Translation Prediction

Jesús González-Rubio
Daniel Ortiz-Martínez
Webinterpret Inc.
{jesus.g,daniel.o}@webinterpret.com

José Miguel Benedí
Francisco Casacuberta
PRHLT Research Center
Univ. Politècnica de València
{jbenedi,fcn}@prhlt.upv.es

Abstract

Current automatic machine translation systems require heavy human proof-reading to produce high-quality translations. We present a new interactive machine translation approach aimed at providing a natural collaboration between humans and translation systems. As such, we grant the user complete freedom to validate and correct any part of the translations suggested by the system. Our approach is then designed according to the requirements placed by this unrestricted proof-reading protocol. In particular, the ability of the system to suggest new translations coherent with the set of potentially disjoint translation segments validated by the user.

We evaluate our approach in a user-simulated setting where reference translations are considered the output desired by a human expert. Results show important reductions in the number of edits in comparison to decoupled post-editing and conventional prefix-based interactive translation prediction. Additionally, we provide evidence that it can also reduce the cognitive overload reported for interactive translation systems in previous user studies.

1 Introduction

Research in the field of *machine translation* (MT) aims at developing computer systems that reduce the effort required to generate translations, whether by assisting human translators or by directly replacing them. However, most research in MT has focused on the development of fully automatic MT approaches. Despite that, except for a handful of very constrained domains, current automatic MT technology still only achieves results that are not satisfactory in practice; automatic MT still require heavy human proof-reading to produce human-quality translations.

We present a new computer-assisted translation approach that integrates human translators and automatic MT into a tight feedback loop. In our approach, the user[1] and the MT system collaborate to generate translations through a series of interactions. At each interaction, the system proposes its best translation for the given input sentence. If the user finds it correct, then it is accepted and the process goes on with the next input sentence. Otherwise, the user makes some corrections that the system takes into account to improve the proposed translation. The rationale behind this *interactive translation prediction* (ITP) approach is to combine the accuracy provided by the human expert with the efficiency of the MT system in contrast to decoupled *post-editing* (PE). Previous works, e.g. (Barrachina et al., 2009), have explored this paradigm; however their practical implementation limits this general proof-reading approach to a prefix-based interaction where the user is forced to correct the errors in the sentence strictly according to the reading order.

Our main contribution, described in Section 3, is a new proof-reading protocol focused on providing a more natural interaction between the user and the system. Specifically, we give complete freedom to the user to validate or to correct any part of the translation at any given interaction. As such, the user is no longer bound to correct the errors following the reading order as in previous prefix-based ITP works (Barrachina et al., 2009; González-Rubio et al., 2013; Green et al., 2014). Preffix-based interaction can be a frustrating and cognitively demanding limitation for the user, and may be a factor in the somehow disappointing

[1] We use the terms "human expert", "human translator", and 'user" indistinctly.

Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL), pages 198–207,
Berlin, Germany, August 7-12, 2016.

results of prefix-based ITP with users (Koehn, 2009; Underwood et al., 2014; Green et al., 2014; Sanchis-Trilles et al., 2014). We design our approach to meet the requirements placed by the unrestricted proof-reading protocol, not the opposite way. The most significant new feature is *conditioned decoding*, for translation generation coherently to a set of segments validated by the user.

An evaluation involving human users is most desirable to study the impact of any proof-reading protocol. However, such a study is expensive, time-consuming and it will require to take into account additional sources of variation, namely the human factor, that may obscure the comparison between different approaches. Therefore, we chose to follow previous works, for instance (Barrachina et al., 2009), and carry out our experiments on a simulated setting intended to provide a direct and, more importantly, objective comparison to previous approaches (Section 4). Regarding evaluation, we propose a new metric to automatically estimate the cognitive load of potential users working on the different ITP environments. To the best of our knowledge, this is the first proposal at this respect. Results in Section 5 confirm the soundness of the proposed ITP approach. Reported figures show important reductions in both the number of corrections typed by the user and her estimated cognitive load.

2 Related Work

Common proof-read MT protocols implement a decoupled PE process in which, first, the MT system returns a translation of a whole given document. Next, a human reads it correcting, in any order, the possible mistakes made by the system.

Interactive approaches (Isabelle and Church, 1998; Langlais and Lapalme, 2002; Tomás and Casacuberta, 2006) were proposed as a more sophisticate way of taking advantage of MT technology. Barrachina et al., (Barrachina et al., 2009) presented a prefix-based ITP approach in which the user is assumed to proof-read each automatic translation correcting each time the first error, if any, in the usual reading order. This can be a reasonable assumption in text or speech transcription (Toselli et al., 2007; Rodríguez et al., 2007) where the output sequence is generated monotonically respect to the input data. However, it has always be an important handicap for translation due to the intrinsic reordering involved in the process.

ITP is a fruitful research field with diverse contributions for multiple authors: (González-Rubio et al., 2010; Alabau et al., 2013; Koehn et al., 2014) among others. We share with (Sanchis-Trilles et al., 2008) the idea of making a more sophisticated use of the mouse actions performed by the user while interacting with the system, and with (González-Rubio et al., 2013) the common ITP formulation for both phrase-based and hierarchical MT models. In particular, we significantly modify the prefix-based ITP implementation presented in the latter work to support the proposed unrestricted proof-reading protocol.

User studies of prefix-based ITP versus PE have shown that while users tend to make less corrections, overall translation time tend to be higher (Koehn, 2009; Underwood et al., 2014; Green et al., 2014; Sanchis-Trilles et al., 2014). Coherently with these results, users also perceive prefix-based ITP as a more cognitive demanding task than PE. This is not surprising given that users are asked to proof-read one new translation (suffix) after each individual correction, which increases significantly the amount of text to be processed to generate a single translation. This is particularly frustrating when the user observes how a correct translation is rewritten with a wrong one by the next suffix suggested by the system. Given that PE do not suffer from this effect, it provides a comprehensive explanation of the somehow disappointing results reported for prefix-based ITP.

To the best of our knowledge, the only alternative to prefix-based proof-reading was proposed in the context of text recognition. Serrano et al., (2014) implement a constrained search procedure that profits from the monotonic alignment between input image, search states and user corrections, to limit the set of possible transcriptions to those coherent with a set of (disjoint) user corrections. We apply a similar idea in a translation context and provide solutions to cope with the non-monotonicity inherent to the task.

3 Beyond Prefix-Based ITP

The goal of our approach is to give complete freedom to the user in her interaction with the system. The process starts when the MT system proposes a full translation of the source language sentence. Then, the user reads the translation and is allowed to validate -all or part of- the correct segments in it and corrects any of its potential errors. Then, the

source (s): No era el hombre más honesto ni el más piadoso , pero era un hombre valiente .
desired translation (t): He was not the most honest or pious of men , but he was courageous .

BEGIN { **MT** : It was not the most honest and the most pious man , but it was a brave man .

IT-1 { **User**: It **was not the most honest** and the most **pious** **of** man **, but** it **was** a brave man **.**
MT : He **was not the most honest** or **pious of** men **, but** it **was** a brave man **.**

IT-2 { **User**: **He was not the most honest or pious of men , but** it **was** **courageous** **.**
MT : **He was not the most honest or pious of men , but** he **was courageous .**

END { **User**: **He was not the most honest or pious of men , but he was courageous .**

Figure 1: Interactive translation of a Spanish sentence into English. First, the system suggests an initial translation. At iteration 1, the user validates the parts of the suggestions she considers to be right and introduces a correction by typing a word: "**of**". This defines a new user feedback with five segments: {"**was not the most honest**", "**pious of**", "**, but**", "**was**", "**.**"}. Then, the system suggests a new translation that contains these segments in the given order. Iteration 2 is similar; the user validates words "**He**" and "**or**", and she types a new correction: "**courageous**". The process ends when the user accepts the translation suggested by the system in the last step. Only two edits are required. In comparison, PE would have needed 10 edits.

system takes into account this feedback to suggest a new translation that contains the segments validated by the user as well as the typed corrections. Such process is repeated until the user validates the whole suggested translation. An example of this process is shown in Figure 1.

The crucial MT feature is the generation of a new translation coherent to the segments already validated by the user. Formally, we represent such user feedback as a sequence of disjoint segments $\mathbf{f} = \tilde{\mathrm{f}}_1, \ldots, \tilde{\mathrm{f}}_k, \ldots, \tilde{\mathrm{f}}_{|\mathbf{f}|}$, where each $\tilde{\mathrm{f}}_k$ is a sequence of consecutive target language words. For example, user feedback at iteration one in Figure 1 is composed of five disjoint segments: $\tilde{\mathrm{f}}_1$ = "was not the most honest", $\tilde{\mathrm{f}}_2$ = "pious of", $\tilde{\mathrm{f}}_3$ = ", but", $\tilde{\mathrm{f}}_4$ = "was" and $\tilde{\mathrm{f}}_5$ = ".". Segments in $\mathbf{f}$ do not overlap and do not necessarily cover the whole sentence. Prefix-based feedback in conventional ITP is a special case of this with only one segment starting at the beginning of the sentence.

Next, we describe the statistical formalization of our approach, the models actually used to implement such formalization, and the search procedures required to efficiently generate translations coherent with this generalized user feedback.

3.1 Statistical Framework

Our problem can be stated as follows: given a source sentence $\mathbf{s} = s_1 \ldots s_{|\mathbf{s}|}$ and some user feedback $\mathbf{f}$, we must find the best target language translation $\mathbf{t} = t_1 \ldots t_{|\mathbf{t}|}$ of $\mathbf{s}$ coherent with $\mathbf{f}$:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} \Pr(\mathbf{t} \mid \mathbf{s}, \mathbf{f})$$

We can make the naïve Bayes' assumption that $\mathbf{s}$ and $\mathbf{f}$ are statistically independent variables given $\mathbf{t}$. This results in the basic equation for ITP with error correction (Ortiz-Martínez, 2011):

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} \Pr(\mathbf{t} \mid \mathbf{s}) \cdot \Pr(\mathbf{f} \mid \mathbf{t}) \qquad (1)$$

where, as we will see in Section 3.2, distribution $\Pr(\mathbf{t} \mid \mathbf{s})$ can be approximated by a machine translation model, and $\Pr(\mathbf{f} \mid \mathbf{t})$ by an error correction model that measures the degree of compatibility between $\mathbf{f}$ and $\mathbf{t}$.

Note that by using a probability distribution $\Pr(\mathbf{f} \mid \mathbf{t})$, any translation is compatible with a given user feedback to some degree. As a consequence, the translation returned by Equation (1) may still not contain the segments validated by the user; we need to identify the sub-string of the returned sentence that corresponds to the each of the segments validated by the user. To solve this problem, we define an alignment $\mathbf{a} = a_1, \ldots, a_{|\mathbf{f}|}$ between the user-validated segments $\mathbf{f} = \tilde{\mathrm{f}}_1, \ldots, \tilde{\mathrm{f}}_{|\mathbf{f}|}$ and a list of segments $\tilde{\mathbf{t}} = \tilde{\mathrm{t}}_1, \ldots, \tilde{\mathrm{t}}_{|\mathbf{f}|}$, where each $\tilde{\mathrm{t}}_k = t_{k_i} \ldots t_{k_j}$ is a sub-sequence of words in $\mathbf{t}$. Each alignment link $a_k = \tilde{\mathrm{t}}_k$ indicates the particular segment in $\mathbf{t}$ that should be replaced by the kth user-validated segment $\tilde{\mathrm{f}}_k$ to make $\mathbf{t}$ coherent to $\mathbf{f}$. Unaligned words in $\mathbf{t}$ constitute the free text that

completes the gaps in between the user-validated segments in $\mathbf{f}$ (González-Rubio et al., 2013). The alignment also must be monotonic to preserve the order of the user-validated segments. Formally, for every pair of alignment links: $a_k = \tilde{\mathrm{t}}_k$ and $a_{k'} = \tilde{\mathrm{t}}_{k'}$, $k < k' \iff k_j < k'_i$.

After including alignment in Equation (1) and following a maximum approximation, we arrive to our final formulation of ITP with error correction:

$$(\hat{\mathbf{t}}, \hat{\mathbf{a}}) = \arg\max_{\mathbf{t},\mathbf{a}} \Pr(\mathbf{t} \mid \mathbf{s}) \cdot \Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{t}) \quad (2)$$

In practice, we combine the probability distributions in Equation (2) in a log-linear fashion as it is typically done in MT (Och and Ney, 2002).

3.2 Models

Equation (2) includes two probability distributions: $\Pr(\mathbf{t} \mid \mathbf{s})$ and $\Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{t})$. The first one can be modeled by any of the multiple machine translation models that have been proposed in the literature; (Koehn, 2009) for example provide a good description of them. We will focus our exposition in the latter distribution, $\Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{t})$, that evaluates the compatibility between a translation $\mathbf{t}$ and some user feedback $\mathbf{f}$ through alignment $\mathbf{a}$.

Following (González-Rubio et al., 2013), we model $\Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{t})$ as an error correction model based on the edit distance (Levenshtein, 1966). Given a candidate string and the corresponding reference string, we model edit distance as a Bernoulli process where each word of the candidate has a probability p_e of being edited. Under this interpretation, the number of edits δ observed in a candidate of length n is a random variable that follows a binomial distribution, $\delta \sim \mathrm{B}(n, p_e)$. By assuming independence between each alignment link, we can model error-correction probability as:

$$\begin{aligned}\Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{t}) &\approx \prod_{k=1}^{|\mathbf{a}|} P_E(\tilde{\mathrm{f}}_k, a_k) \\ &= \prod_{k=1}^{|\mathbf{a}|} \binom{n_k}{\delta_k} p_e^{\delta_k} (1 - p_e)^{(n_k - \delta_k)}\end{aligned}$$

where $P_E(\tilde{\mathrm{f}}_k, a_k)$ is the error correction probability for the k-th alignment link whose value is given by the probability mass function of the binomial distribution, $n_k = |\tilde{\mathrm{f}}_k|$ is the length in words of the k-th segment validated by the user ($\tilde{\mathrm{f}}_k$), and δ_k is the edit distance between $\tilde{\mathrm{f}}_k$ and the segment $a_k = \tilde{\mathrm{t}}_k$ of $\mathbf{t}$ aligned to it according to $\mathbf{a}$.

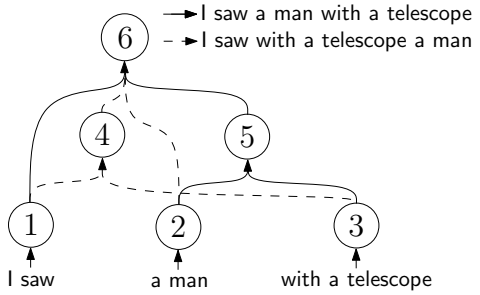

Figure 2: Example of a hypergraph encoding two different translations for the Spanish sentence: "Vi a un hombre con un telescopio".

The probability of editing p_e is the single free parameter of this model. Alternatively, we can use a model based on a multinomial distribution assigning different probabilities to different edit operations. Nevertheless, we adhere to the binomial approximation due to its simplicity.

3.3 Search

Next, we address the problem posed by the maximization in Equation (2). Following (Barrachina et al., 2009), we split search into a two step process. Given a source language sentence, we first generate a graph-based representation that contains its most probable translations. Then, we search for the optimal translation and alignment on it according to Equation (2). In particular, we use *hypergraphs* to represent such search space.

One important advantage of this approach, is that it separates the proof-read step from the MT engine used to generate the initial translations. As such it provides an unified framework that accepts both the use of phrase-based and hierarchical/syntax translation models.

3.3.1 Hypergraphs

A hypergraph (Gallo et al., 1993) is a generalization of the concept of graph where the edges (now called hyperedges) may connect several nodes (hypernodes) at the same time. Formally, a hypergraph is a weighted acyclic graph represented by a pair $\mathcal{H} = <\mathcal{V}, \mathcal{E}>$, where $\mathcal{V}$ is a set of hypernodes and $\mathcal{E}$ is a set of hyperedges. Each hyperedge $\varepsilon \in \mathcal{E}$ connects a set of tail hypernodes $\mathcal{T}(\varepsilon) = \{\tau_1 \dots \tau_{|\mathcal{T}(\varepsilon)|}\}$ $\tau_l \in \mathcal{V}$, to a head hypernode $\mathrm{H}(\varepsilon) \in \mathcal{V}$. A hypernode with no ingoing hyperedges is a leaf, while a hypernode with no outgoing hyperedges is a root. Each hypernode represents a partial translation generated during the MT decoding process. Each ingoing hyperedge ε

represents the rule applied to generate the partial solution in the head from the partial solutions in the tail hypernodes, as such it has an associated probability $P(\varepsilon)$. Figure 2 shows an example hypergraph[2]. Two alternative translations are constructed from the leaf hypernodes (1, 2 and 3) up to the root hypernode (6). Hypergraphs provide a compact representation of the translation space that allows us to derive efficient search algorithms.

Hypergraphs are the natural representation for hierarchical MT models (Chiang, 2005; Zollmann and Venugopal, 2006). Note, however, that word-graphs (Ueffing et al., 2002), which are used to represent the search space for phrase-based models (Koehn et al., 2003), are a special case of hypergraphs in which hyperedges have at most one tail hypernode.

3.3.2 Search on hypergraphs

We formalize the maximization in Equation (2) as a bottom-up search problem. Starting from the leaf hypernodes, we keep track of the best solutions (partial translation and alignment) achievable at each hypernode. We define $\mathrm{Q}(\nu, [m, n])$ as the probability of the most likely partial translation derivable from hypernode ν aligned to (accounting for) user-validated segments from m-th to n-th, we will refer to this interval as the coverage of the partial solution. Given a node ν and a coverage, we compute its score from its ingoing hyperedges. Specifically, $\mathrm{Q}(\nu, [m, n])$ will be equal to the maximum score of the partial solutions computed from any ingoing hyperedge. Partial solutions from an ingoing hyperedge ε are defined as combinations of partial solutions on its tail hypernodes under the constrain that the concatenation of their coverages equals $[m, n]$. Formally, given an ingoing hyperedge ε, a combination $\mathbf{c} = \{\mathrm{Q}(\tau_l, [m_l, n_l])\}_{l=1}^{|\mathcal{T}(\varepsilon)|}$ is valid if it holds the following four conditions:

1. $m_1 = m$
2. $n_{|\mathcal{T}(\varepsilon)|} = n$
3. $\forall l, \tau_l \in \mathcal{T}(\varepsilon)$
4. $\forall l : 1 < l \leq |\mathcal{T}(\varepsilon)|, m_l = n_{l-1} + 1$

$\mathrm{Q}(\nu, [m, n])$ can be computed efficiently via the following dynamic programming recursion:

$$\mathrm{Q}(\nu, [m, n]) = \max_{\substack{\varepsilon \in \mathcal{I}(\nu) \\ \mathbf{c} \in \mathcal{C}(\varepsilon, [m,n])}} P(\varepsilon) \prod_{\mathrm{Q}(\tau_l, [m_l, n_l]) \in \mathbf{c}} \mathrm{Q}(\tau_l, [m_l, n_l])$$

[2]For simplicity, we do not show hyperedge probabilities.

	EU corpus (Es/En)		
	train	tune	test
Sentences	214K	400	800
Tokens	5.9M / 5.2M	12K / 10K	23K / 20K
Vocabulary	97K / 84K	3K / 3K	5K / 4K

Table 1: Main figures of the EU corpus. K and M stand for thousands and millions of elements.

where $\mathcal{I}(\nu)$ are the ingoing hyperedges of ν, $P(\varepsilon)$ is the probability of hyperedge ε, $\mathcal{C}(\varepsilon, [m, n])$ is the set of valid combinations for hyperedge ε and coverage $[m, n]$, and $\mathbf{c} \in \mathcal{C}(\varepsilon, [m, n])$ is one of such valid combinations

Leaf hypernodes represent the base cases for this recursion. For simplicity, we restrict them to be fully-aligned to at most one user-validated segment[3]. That is, given a leaf hypernode $\lambda \in \mathcal{V}$:

$$\mathrm{Q}(\lambda, [m, n]) = \begin{cases} P_{\mathrm{MT}}(\lambda) P_E(\mathrm{w}(\lambda), \tilde{\mathrm{f}}_m) & \text{if } m = n \\ 0 & \text{otherwise.} \end{cases}$$

where $P_{\mathrm{MT}}(\lambda)$ is the MT probability (language plus translation model) of λ, and $P_E(\mathrm{w}(\lambda), \tilde{\mathrm{f}}_m)$ is the error correction probability between the target text covered by the leaf hypernode, $\mathrm{w}(\lambda)$, and the m-th user-validated segment in $\mathbf{f}$.

The score of the optimal solution is given by $\mathrm{Q}(\alpha, [1, |\mathbf{f}|])$, where $\alpha \in \mathcal{V}$ is the root hypernode. We can recover $(\hat{\mathbf{t}}, \hat{\mathbf{a}})$ through backtracking.

The process described above loops over all hyperedges and coverages (bounded by $|\mathcal{E}||\mathbf{f}|^2$), evaluating all valid combinations (bounded by the coverage partitions). It can be implemented by an algorithm with a complexity in $O(|\mathcal{E}||\mathbf{f}|^{2\tau})$, where τ is the average number of tail hypernodes per hyperedge (usually set to 2). In practice, our approach has a complexity in $O(|\mathcal{E}||\mathbf{f}|^4)$.

4 Experimental Setup

4.1 Corpus and MT systems

We tested the proposed methods in the Spanish-to-English (Es–En) partition of the *Bulletin of the European Union* (EU) corpus (Barrachina et al., 2009; González-Rubio et al., 2013). We tokenized the corpus keeping the real case of the sentences. Table 1 shows the main figures of the corpus.

We estimated a hierarchical MT model for the train partition with the standard configuration of

[3]Preliminary experiments did not show a difference in the final results when relaxing this restriction.

the Moses toolkit (Koehn et al., 2007). Log-linear weighs were estimated by minimum error-rate training (Och, 2003) on the tune partition. Then, we automatically translated tune and test partitions using the optimized model to obtain the corresponding hypergraphs. Next, we optimized the single free parameter p_e of the error correction model (see Section 3.2) on the tune partition. Finally, we interactively translated both partitions according to the unrestricted ITP approach proposed in Section 3.

4.2 User Simulation

ITP evaluation with human translators is simply to slow and expensive to be applied on a frequent and ongoing basis during system development. Instead, we carried out an automatic evaluation with simulated users which is faster and cheaper.

At each ITP iteration (see Figure 1), we have to decide which segments in the suggested translation should be validated, and which error should be corrected. To do that, we considered the reference translations in the corpus as the output that a human expert would want to obtain. Then, we align the suggested translation and the reference via edit distance: words aligned to itself are marked as valid, while edited words are potential corrections to be typed by the simulated user.

Without loss of generality, we introduced two restrictions: (1) we restrict users to validate segments only at the first iteration, and (2) the simulated user always corrected the first (in reading order) error in the suggested translation. We are aware that the results obtained with this user simulation will be pessimistic since it forbids behaviors that may improve user productivity, e.g. validating segments at each iteration or correcting more promising parts of the suggested translation. Our goal is not to match the behavior of a human translator, but to allow for a meaningful comparison against conventional ITP. Note that prefix-based proof-reading is a particular case of our user simulation with no segment validation.

4.3 Evaluation Metrics

ITP systems are evaluated according to the effort needed to generate the desired translations. This effort is usually estimated as the number of actions performed by the user while interacting with the system. In our user simulation, we describe two different actions: segment-validation, and word-correction. Each segment validation involves the user to "click" on the initial and final words of the segment[4]. Each correction corresponds to an edit operation performed by the user. Specifically, we used the following measures in our experiments:

Word stroke ratio (WSR): Proposed in (Tomás and Casacuberta, 2006) as the quotient between the number of words edited by the user (word-strokes), and the number of words in the final translation. Word-strokes are considered as single actions with constant cost independently of the length of the edited word.

Mouse action ratio (MAR): Proposed in (Barrachina et al., 2009) as the quotient between the number of "clicks" made by the user (mouse-actions), and the number of words in the final translation. In addition to the "clicks" for segment validation, we count one more mouse action per sentence accounting for the final acceptance of the suggested translation.

Conceptually (Macklovitch et al., 2005), MAR can be seen as accounting for the cognitive part of the supervision process: undertanding the translation and identifying the errors in it, while WSR accounts for the actual physical effort required to type the corrections. As such, both metrics are complementary to express the total human effort involved in proof-reading a document.

We also evaluated the quality of the initial automatic translations generated by the system:

Bilingual evaluation understudy (BLEU): Proposed in (Papineni et al., 2002), it is based on the precision of n-grams between the suggested translation and the reference; it also includes a *brevity penalty* to penalize short translations. This score ranges between 0 and 100, with 100 denoting a perfect translation.

Translation edit rate (TER): Proposed in (Snover et al., 2006), it measures the number of edit operations (substitution, insertion and deletion of single words, and swap of word sequences) divided by the number of words in the reference.

In addition to be an MT quality metric, TER can also be seen as a human-effort measure in PE scenarios. Therefore, we can use TER and WSR to compare human effort between PE and ITP.

[4]One single "click" is enough for one-word segments.

AUTOMATIC TRANSLATION

	tune		test	
	BLEU [%](⇑)	TER [%](⇓)	BLEU [%](⇑)	TER [%](⇓)
	38.9±1.4	47.2±1.4	44.2±1.3	41.1±1.2

COMPUTER-ASSISTED TRANSLATION

	TER [%](⇓)		TER [%](⇓)	
Post-editing	47.2±1.4		41.1±1.2	
	MAR [%](⇓)	WSR [%](⇓)	MAR [%](⇓)	WSR [%](⇓)
prefix-based ITP	11.2±0.4	45.8±2.0	10.3±0.3	54.5±1.4
Our approach	33.9±1.2	**30.5±1.6**	35.4±0.9	**35.1±1.1**

Table 2: Results of different approaches when interactively translating the tune and test partitions of the EU corpus. We compare decoupled PE, prefix-based ITP and the unrestricted ITP approach proposed in this work. Automatic translation results are shown to indicate the difficulty of the task. Results in bold indicate the lowest human effort (typing) achievable by the different scenarios.

Finally, in order to assess the statistical significance of the results, we also provide 95% confidence intervals for their values. These intervals were computed via pair-wise bootstrap resampling as proposed in (Zhang and Vogel, 2004).

5 Results

This section presents the results of the experiments performed to assess the unrestricted ITP approach proposed in Section 3. First, we compare our ITP approach to the prefix-based ITP scenario described in (Barrachina et al., 2009) and the decoupled PE approach. Then, we further study our approach investigating the relationship between segment-validation and typing effort. Finally, we provide evidence that the proposed unrestricted proof-reading protocol allows to reduce the cognitive overload produced by the changing translation completions of prefix-based ITP approaches.

Table 2 displays user-effort results for the proposed ITP approach against prefix-based ITP (Barrachina et al., 2009)[5] and a decoupled PE baseline approach. Automatic translation results are also displayed to give an idea of the difficulty of the task. We can observe how our approach clearly outperformed both prefix-based ITP and PE in terms of user typing effort as measured by WSR and TER respectively. According to these results, a human translator assisted by our ITP system would only need to correct only about one third of the words to generate the correct translations. In comparison, PE would require to type

[5]For prefix-based ITP, MAR accounts for the prefixes validated by the user while proof-reading the translations.

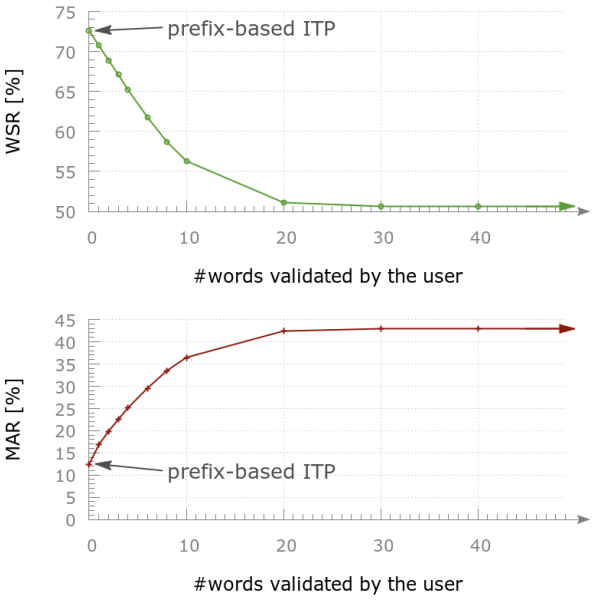

Figure 3: WSR and MAR as a function of the maximum number of words validated by the simulated user in our ITP approach. No changes were observed for more than 40 validated words.

$\sim 41\%$ of the words (17% more) while prefix-based ITP would require to correct more that half of them (55% more). Additionally, we can observe that prefix-based ITP was not better than the PE baseline in all cases. This result, coherent with previous works e.g. (González-Rubio et al., 2013; Green et al., 2014), exemplifies the potential limitations of prefix-based ITP.

The large reductions in typing effort observed for our ITP approach came together with an important increase in the number of mouse actions.

Next, we focused on the differences between prefix-based and our approach. To do that, we carried out different experiments allowing the simulated user to validate an increasing number of words from zero to infinity (corresponding respectively to the results of prefix-based ITP and our approach in Table 2). Figure 3 shows WSR (top) and MAR (bottom) for the Test partition as a function of the maximum number or words allowed to be validated by the user.

As we allowed the simulated user to validate more words, the amount of words to be corrected (WSR) decreased dramatically. For example, we obtained a 10% relative reduction in WSR when we allowed the user to validate a maximum of 4 words. A similar trend (but in the opposite direction) can be observed for MAR: as we allowed the user to validate more words, the number of mouse actions increased until stabilization. In other words, our ITP approach allows to reduce user typing effort at the expense of an increase in the number of mouse actions. As we have said before, WSR and MAR account for different phenomena and thus have different cost from a human point of view (Macklovitch et al., 2005). It may seem that we have simply exchanged typing effort for cognitive effort. However, two considerations allow us to consider this a beneficial exchange. On the one hand, from a pure mechanistic point of view, typying a whole word usually requires more effort than "clicking" on it. On the other hand, from a cognitive point of view, the user has to read, understand, and evaluate the suggested translation in both prefix-based ITP and our approach. Hence, the difference in cognitive effort between these two approaches approaches is most probably negligible. Nevertheless, these considerations should be tested with actual human users before reaching categorical conclusions.

Average response time of our Python prototype was below 3 seconds[6]. Obviously, it does not qualify as real-time. However, we expect an important reduction in response time after implementing our approach in a more efficient language.

We performed a final analysis to evaluate to which extent our proposal alleviates the main annoying effect inherent to prefix-based ITP, namely correct words in a given suffix overwritten by the next suggested suffix. This common effect, observed in several user studies, make human users

[6]The test machine was an Intel i5 CPU at 3.4 GHz.

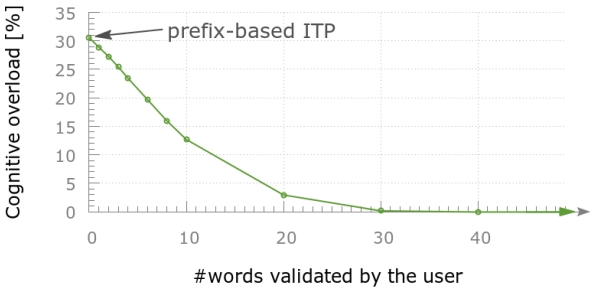

Figure 4: Percentage of words suggested by the system that were correct but overwritten in subsequent translation suggestions, as a function of the number of words validated by the simulated user. This ratio can be seen as a measure of the user cognitive overload.

feel that cognitive effort invested in evaluating each suggested translation was wasted.

To do that, we measured the number of correct words that were modified by subsequent translation suggestions, normalized by the total number of suggested words. Figure 4 displays this percentage as a function of the maximum number of words that can be validated by the simulated user. We can observe how for prefix-based ITP (zero value in the x-axis), this cognitive overload is a very important phenomena; more than 30% of the words suggested by the system were correct but modified by following suffix suggestions. As we allowed more words to be fixed, this percentage steadily decreased down to zero. This indicates that our ITP proposal actually provides a mechanism to overcome the cognitive overload inherent to prefix-based ITP.

6 Summary

We have presented a new ITP approach where the user is not longer bound to interact with the system in a prefix-based fashion (Barrachina et al., 2009). The proposed ITP approach gives the user complete freedom to validate and correct any part of the suggested translations thus providing a more natural working environment for human translators. We formalize the problem as a MT model with error correction which, in practice, is implemented as a constrained search on hypergraphs.

Simulated results showed that the proposed ITP approach drastically reduced the typing effort needed to generate translations, improving results of both decoupled PE and prefix-based ITP. This

reduction in typing effort came at the expense of a larger amount of mouse actions required to validate correct segments of the suggested translations. However, since mouse actions are cheaper than typing full words, we can expect this exchange to reduce overall user effort. Nevertheless, this expectation should be confirmed in future experiments with actual human users. Finally, in addition to reduce user effort, we also provide evidence indicating that the proposed ITP approach can reduce the cognitive overload commonly reported by humans using prefix-based ITP systems.

Acknowledgments

Work supported by the Generalitat Valenciana under grant ALMAMATER (PrometeoII/2014/030).

References

Vicent Alabau, Jesús Gonzlez-Rubio, Luis A. Leiva, Daniel Ortiz-Martínez, Germán Sanchis-Trilles, Francisco Casacuberta, Bartolomé' Mesa-Lao, Ragnar Bonk, Michael Carl, and Mercedes García-Martínez. 2013. User evaluation of advanced interaction features for a computer-assisted translation workbench. In *Proceedings of the XIV Machine Translation Summit*, pages 361–368.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35:3–28, March.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 263–270.

Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. 1993. Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42(2-3):177–201, April.

Jesús González-Rubio, Daniel Ortiz-Martínez, and Francisco Casacuberta. 2010. Balancing user effort and translation error in interactive machine translation via confidence measures. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 173–177.

Jesús González-Rubio, Daniel Ortíz-Martinez, José-Miguel Benedí, and Francisco Casacuberta. 2013. Interactive machine translation using hierarchical translation models. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 244–254.

Spence Green, Sida I. Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D. Manning. 2014. Human effort and machine learnability in computer aided translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 1225–1236.

Pierre Isabelle and Ken Church. 1998. *Special issue on: New tools for human translators*, volume 12. Kluwer Academic Publishers, January.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

Philipp Koehn, Chara Tsoukala, and Herve Saint-Amand. 2014. Refinements to interactive translation prediction based on search graphs. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 574–578.

Philipp Koehn. 2009. A process study of computer-aided translation. *Machine Translation*, 23(4):241–263.

Philippe Langlais and Guy Lapalme. 2002. TransType: development-evaluation cycles to boost translator's productivity. *Machine Translation*, 17(2):77–98, September.

Vladimir Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, February.

Elliot Macklovitch, Nam-Trung Nguyen, and Roberto Silva. 2005. User evaluation report. Technical report, Université de Montréal. TransType2 (IST-2001-32091).

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302.

Franz Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167.

Daniel Ortiz-Martínez. 2011. *Advances in Fully-Automatic and Interactive Phrase-Based Statistical Machine Translation*. Ph.D. thesis, Universitat Politècnica de València.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

Luis Rodríguez, Francisco Casacuberta, and Enrique Vidal. 2007. Computer assisted transcription of speech. In *Proceedings of the 3rd Iberian Conference on Pattern Recognition and Image Analysis*, volume 4477 of *Lecture Notes in Computer Science*, pages 241–248.

Germán Sanchis-Trilles, Daniel Ortiz-Martínez, Jorge Civera, Francisco Casacuberta, Enrique Vidal, and Hieu Hoang. 2008. Improving Interactive Machine Translation via Mouse Actions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 485–494.

Germán Sanchis-Trilles, Vicent Alabau, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin L. Hill, Philipp Koehn, Luis A. Leiva, Bartolomé Mesa-Lao, Daniel Ortiz-Martínez, Herve Saint-Amand, and Chara Tsoukala. 2014. Interactive translation prediction versus conventional post-editing in practice: a study with the casmacat workbench. *Machine Translation*, 28(3-4):217–235, December.

Nicols Serrano, Adri Gimnez, Jorge Civera, Alberto Sanchis, and Alfons Juan. 2014. Interactive handwriting recognition with limited user effort. *International Journal on Document Analysis and Recognition*, 17:47–59.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Jesús Tomás and Francisco Casacuberta. 2006. Statistical phrase-based models for interactive computer-assisted translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 835–841.

Alejando H. Toselli, Verónica Romero, Luis Rodriguez, and Eenrique Vidal. 2007. Computer assisted transcription of handwritten text images. In *9th International Conference on Document Analysis and Recognition*, volume 2, pages 944–948.

Nicola Ueffing, Franz J. Och, and Hermann Ney. 2002. Generation of word graphs in statistical machine translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 156–163.

Nancy Underwood, Bartolomé Mesa-Lao, Mercedes García Martinez, Michael Carl, Vicent Alabau, Jesús González-Rubio, Luis A. Leiva, Germán Sanchis-Trilles, Daniel Ortíz-Martinez, and Francisco Casacuberta. 2014. Evaluating the effects of interactivity in a post-editing workbench. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 553–559.

Ying Zhang and Stephan Vogel. 2004. Measuring confidence intervals for the machine translation evaluation metrics. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 85–94.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 138–141.