Domingo-Ballester, M.; Peris-Abril, Á.; Casacuberta Nolla, F. (2017). Segment-based interactive-predictive machine translation. Machine Translation. 31(4):163-185. doi:10.1007/s10590-017-9213-3

Noname manuscript No.
(will be inserted by the editor)

Segment-Based Interactive-Predictive Machine Translation

Miguel Domingo · Álvaro Peris · Francisco Casacuberta

Abstract Machine translation systems require human revision to obtain high-quality translations. Interactive methods provide an efficient human–computer collaboration, notably increasing productivity. Recently, new interactive protocols have been proposed, seeking for a more effective user interaction with the system. In this work, we present one of these new protocols, which allows the user to validate all correct word sequences in a translation hypothesis. Thus, the left-to-right barrier from most of the existing protocols is broken. We compare this protocol against the classical prefix-based approach, obtaining a significant reduction of the user effort in a simulated environment. Additionally, we experiment with the use of confidence measures to select the word the user should correct at each iteration, reaching the conclusion that the order in which words are corrected does not affect the overall effort.

Keywords Machine Translation, Computer-assisted Translation, Interactive-predictive Machine Translation

1 Introduction

Despite obtaining admissible results in many tasks, machine translation (MT) technology is still far from automatically obtaining high-quality translations (Dale, 2016). To cope with this problem, a human agent needs to supervise the MT hypotheses in a post-editing stage. This supervision process makes for a more efficient working method than a completely manual translation. However, higher efficiency rates can be achieved if human and computer work together on a joint strategy. With the goal of combining the knowledge of a human translator and the efficiency of an MT system, the so-called interactive-predictive machine translation (IMT) was introduced within the *TransType* project (Foster et al., 1997), and was further developed in the TransType2 (Barrachina et al., 2009) and CasMaCat (Alabau et al., 2013) projects.

This approach is an iterative process in which the user corrects the leftmost wrong word from the hypothesis generated by the system. This correction, together with the previous words, conforms a validated prefix. At each new iteration, the system generates a suffix that completes the prefix to produce a new translation hypothesis. Therefore, the user steadily validates a larger prefix, until the system hypothesis corresponds to the desired translation.

During the last years, IMT has been an active research field, and many novelties were introduced. Different contributions to the generation of the new suffix were developed (Koehn et al., 2014; Torregrosa et al., 2014; Azadi and Khadivi, 2015). González-Rubio et al. (2010) added confidence measures to assist the user to validate new prefixes. Sanchis-Trilles et al. (2008) profited from the use of the mouse for validating a prefix and suggesting a new suffix each time the user clicked on a position to type a word. Alabau et al. (2011) and Alabau et al. (2014) introduced multimodal

Miguel Domingo · Álvaro Peris · Francisco Casacuberta
Pattern Recognition and Human Language Technology Research Center, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain,
E-mail: midobal@prhlt.upv.es, lvapeab@prhlt.upv.es, fcn@prhlt.upv.es

interaction into the IMT environment, integrating handwriting and speech recognition. Online learning was also used for improving the system with the user feedback (Nepveu et al., 2004; Ortiz-Martínez, 2016). Marie and Max (2015) introduced a touch-based interaction to iteratively improve translation quality. Cheng et al. (2016) presented a new framework in which, at each iteration, the user corrected the most critical error from the translation hypothesis. Recently, the interactive framework has also been deployed for the novel neural machine translation approach (Knowles and Koehn, 2016; Wuebker et al., 2016; Peris et al., 2017). However, the core of the user protocol remained the same in most of these works.

This prefix-based protocol presents a cumbersome phenomenon when the non-validated part of the sentence contains correct words: if the system modifies those words in following predictions, the user must correct words which were already correct in previous iterations. This results in an increase of the user effort, as well as in an annoying system behavior.

To overcome this weakness, new protocols that allow the user to validate all correct sub-strings of a translation hypothesis were recently proposed (Domingo et al., 2016; González-Rubio et al., 2016; Peris et al., 2017). In this work, we present a simplified version of one of these protocols. The proposed protocol makes use of some features of `Moses` (Koehn et al., 2007), a widely use toolkit for statistical machine translation (SMT) (Koehn, 2010).

Our proposal shares some similarities with Marie and Max (2015) in the sense that we select word segments from a translation hypothesis. However, on the one hand, our protocol considers more types of user interactions such as word corrections and word deletions (see Section 2.2). On the other hand, we have different goals in mind: Marie and Max (2015) aim to increase translation quality with the help of a human user, who selects the correct parts of a translation hypothesis. We aim to reduce the human effort when generating the highest quality translation. Our main contributions are the following:

1. We formally present the segment-based protocol proposed by Domingo et al. (2016).
2. We present a simple implementation of the segment-based protocol. This implementation takes advantage of the SMT toolkit `Moses`.
3. We conduct more experiments to compare this new protocol against classical IMT. Such experimentation include more language pairs and larger translation tasks.
4. We conduct experiments using confidence measures.

The rest of this paper is structured as follows: Section 2 describes the main concepts and statistical formalization of IMT. Then, in Section 3, we report the experiments conducted in order to assess our proposal. After that, in Section 4, we show and discuss the experimental results. Finally, conclusions of the work are drawn in Section 5.

2 Interactive Machine Translation

Classical IMT approaches (Barrachina et al., 2009; Alabau et al., 2013) are based on the statistical formalization of the MT problem. Given a source sentence $\mathbf{x}$, the goal of SMT is to find the best translation $\hat{\mathbf{y}}$ (Brown et al., 1993):

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} Pr(\mathbf{y} \mid \mathbf{x}) \tag{1}$$

This expression is usually approximated by means of the so-called phrase-based models (Koehn, 2010). They rely on a log-linear combination of different models (Och and Ney, 2002); namely, phrase-based alignment models, reordering models and language models, among others (Zens et al., 2002; Koehn et al., 2003). The search is usually performed employing a stack decoding algorithm. However, it is worth mentioning the great impact that neural machine translation (Sutskever et al., 2014; Bahdanau et al., 2015) had in the last few years. This is a novel and competitive technology, based on the sole use of neural networks for carrying out the translation process.

2.1 Prefix-Based Interactive Machine Translation

In the prefix-based IMT approach, the user–computer collaboration starts with the system proposing an initial translation $\mathbf{y}$ of length I. Then, the user searches for the leftmost wrong word y_i and corrects it. With this action, all preceding words are inherently validated, forming a validated prefix $\tilde{\mathbf{y}}_p$, that includes the corrected word $\tilde{y}_i$. The system then reacts to this user feedback, generating a suffix $\hat{\mathbf{y}}_s$ that completes $\tilde{\mathbf{y}}_p$, to obtain a new translation of $\mathbf{x}$: $\hat{\mathbf{y}} = \tilde{\mathbf{y}}_p \, \hat{\mathbf{y}}_s$. This process is repeated until the user accepts the system complete suggestion. Fig. 1 shows an example of a prefix-based IMT session.

source ($\mathbf{x}$): Si vous avez été exposé , vous devriez consulter votre médecin pour des tests
target translation ($\hat{\mathbf{y}}$): If you have been exposed , you should go to your doctor for tests

IT-0	**MT**	If you have been exposed , you should consult your doctor for tests
IT-1	**User**	If you have been exposed , you should **go** your doctor for tests
	MT	If you have been exposed , you should go consult your doctor for tests
IT-2	**User**	If you have been exposed , you should go **to** your doctor for tests
	MT	If you have been exposed , you should go to consult your doctor for tests
IT-3	**User**	If you have been exposed , you should go to **your** your doctor for tests
	MT	If you have been exposed , you should go to your doctor for tests
END	**User**	If you have been exposed , you should go to your doctor for tests

Fig. 1 Prefix-based IMT session to translate a French sentence into English. At the initial iteration (*IT-0*), the system suggests an initial translation. Then, at iteration 1, the user corrects the leftmost wrong word (**go**). With this action, the user is inherently validating the prefix If you have been exposed , you should . Taking this user feedback into account, the system suggests a new hypothesis. Similarly, at iteration 2, the user corrects the leftmost wrong word (**to**). The session ends when the user accepts the last translation suggested by the system.

The suffix generation was formalized by Barrachina et al. (2009) as follows:

$$\hat{\mathbf{y}}_s = \arg\max_{\mathbf{y}_s} Pr(\mathbf{y}_s \mid \mathbf{x}, \tilde{\mathbf{y}}_p) \tag{2}$$

which can be straightforwardly rewritten as:

$$\hat{\mathbf{y}}_s = \arg\max_{\mathbf{y}_s} Pr(\tilde{\mathbf{y}}_p \, \mathbf{y}_s \mid \mathbf{x}) \tag{3}$$

This equation is very similar to Eq. 1: at each iteration, the process consists in a regular search in the translations space but constrained by the prefix $\tilde{\mathbf{y}}_p$.

2.2 Segment-Based Interactive Machine Translation

The segment-based IMT approach extends the human–computer collaboration. Now, at each iteration, the user can validate segments (sequences of words), delete all the words between two segments (if any) to create a larger segment or correct a word. Fig. 2 shows an example of an IMT session using this approach.

As in the prefix-based approach, the process starts with the system suggesting an initial translation. Then, the user searches for those sequences of words which she considers that are correct, and validates them. After that, she can delete words between validated segments, in order to create a larger segment. Finally, the user corrects a word. Fig. 3 exemplifies the possible user actions.

source ($\mathbf{x}$): Si vous avez été exposé , vous devriez consulter votre médecin pour des tests
target translation ($\hat{\mathbf{y}}$): If you have been exposed , you should go to your doctor for tests

IT-0	**MT**	If you have been exposed , you should consult your doctor for tests
IT-1	**User**	If you have been exposed , you should **go** your doctor for tests
	MT	If you have been exposed , you should consult go your doctor for tests
IT-2	**User**	If you have been exposed , you should go **to** your doctor for tests
	MT	If you have been exposed , you should go to your doctor for tests
END	**User**	If you have been exposed , you should go to your doctor for tests

Fig. 2 Segment-based IMT session to translate a French sentence into English. At the initial iteration (*IT-0*), the system suggests an initial translation. Then, at iteration 1, the user validates those segments which are correct (if you have been exposed , you should , and your doctor for tests) and types a word correction (**go**). With this information, the system suggests a new hypothesis. At iteration 2, the user deletes a word (*consult*) to create a larger segment (if you have been exposed , you should go) and types a new word correction (**to**). The session ends when the user accepts the last translation suggested by the system.

Reference: If you have been exposed , you should go to your doctor for tests
Hypothesis: If you have been exposed , you should consult go your doctor for tests

Segment validation: If you have been exposed , you should consult go your doctor for tests

Words deletion: If you have been exposed , you should ~~consult~~ go your doctor for tests

Word correction: If you have been exposed , you should go **to** your doctor for tests

Fig. 3 Example of the possible user actions in segment-based IMT. The example corresponds to iteration 2 from Fig. 2. To make the example more illustrative, we consider as if this were the first iteration and segments had not yet been validated. First, the user validates the correct word sequences (If you have been exposed , you should , go and your doctor for tests). Then, she deletes some words (~~consult~~) to create a bigger segment (If you have been exposed , you should go). Finally, the user corrects a word (**to** is added between two validated segments).

These three actions constitute the user feedback that is inputted to the system, as part of the interactive process. This feedback has the form $\tilde{\mathbf{f}}_1^N = \tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_N$, where $\tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_N$ is the sequence of N correct segments validated by the user in an interaction. Each segment is defined as a sequence of one or more target language words. Therefore, each action taken by the user modifies the feedback in a different way. Thus, the user can:

1. Validate a new segment, inserting a new segment $\tilde{\mathbf{f}}_i$ in $\tilde{\mathbf{f}}_1^N$.
2. Delete words between two segments, merging two consecutive segments $\tilde{\mathbf{f}}_i, \tilde{\mathbf{f}}_{i+1}$ into a new one.
3. Introduce a word correction. This is introduced as a new one-word validated segment, $\tilde{\mathbf{f}}_i$, which is inserted in $\tilde{\mathbf{f}}_1^N$.

The first two actions are optional: at a given iteration, the user might not validate new segments or delete words. The last action is mandatory. Once a new word correction is introduced, the system reacts to the user feedback, starting a new iteration of the process.

The system reacts to this feedback generating a sequence of new translation segments $\hat{\mathbf{h}}_0^{N+1} = \hat{\mathbf{h}}_0, \ldots, \hat{\mathbf{h}}_{N+1}$. That means, an $\hat{\mathbf{h}}_i$ for each pair of validated segments $\tilde{\mathbf{f}}_i, \tilde{\mathbf{f}}_{i+1}$, being $1 \leq i \leq N$; plus one more at the beginning of the hypothesis, $\hat{\mathbf{h}}_0$; and another at the end of the hypothesis, $\hat{\mathbf{h}}_{N+1}$. The new translation of $\mathbf{x}$ is obtained by alternating validated and non-validated segments: $\hat{\mathbf{y}} = \hat{\mathbf{h}}_0, \tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_N, \hat{\mathbf{h}}_{N+1}$. We want to obtain the best sequence of translation segments, given the feedback and the source sentence:

$$\hat{\mathbf{h}}_0^{N+1} = \arg\max_{\mathbf{h}_0^{N+1}} Pr(\mathbf{h}_0^{N+1} \mid \mathbf{x}, \tilde{\mathbf{f}}_1^N) \tag{4}$$

which can be rewritten as:

$$\hat{\mathbf{h}}_0^{N+1} = \arg\max_{\mathbf{h}_0^{N+1}} Pr(\mathbf{h}_0, \tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_N, \mathbf{h}_{N+1} \mid \mathbf{x}) \quad (5)$$

This last equation is very similar to the classical prefix-based IMT equation (Eq. 2). Now, the search is performed in the space of possible substrings of the translations of $\mathbf{x}$, constrained by the sequence of segments $\tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_N$, instead of being limited to the space of suffixes constrained by $\tilde{\mathbf{y}}_p$, as in Eq. 2.

3 Experiments

In this section, we present the experiments conducted for assessing our proposal. We also describe the corpora and set up our experimental framework.

3.1 Corpora

Following prior IMT works (Tomás and Casacuberta, 2006; Barrachina et al., 2009), we tested our proposal with five different corpora: The EMEA corpus[1] (Tiedemann, 2009), which is formed by medical documents from the *European Medical Agency*. The EU corpus (Barrachina et al., 2009), which was extracted from the *Bulletin of the European Union*. TED[2] (Federico et al., 2011), which is a collection of public speeches from a variety of topics. Xerox (Barrachina et al., 2009), which was created from *Xerox* printer manuals. And Europarl[3][4] (Koehn, 2005), which is a collection of proceedings from the European Parliament.

All datasets were kept truecased, except for the Chinese–English language pair from TED, since Chinese has no case information. All datasets were tokenized using the standard tool provided by the `Moses` (Koehn et al., 2007) toolkit. Chinese sentences were split into words using the Standford word segmenter (Tseng et al., 2005). Table 1 shows the main features of the corpora.

Table 1 Corpora statistics. K denotes thousands and M millions. $|S|$ stands for number of sentences, $|T|$ for number of tokens and $|V|$ for size of the vocabulary. ***Fr*** denotes French; ***En***, English; ***De***, German; ***Es***, Spanish and ***Zh***, Chinese.

Corpus	Languages	Train			Development			Test		
		$\|S\|$ *(K)*	$\|T\|$ *(M)*	$\|V\|$ *(K)*	$\|S\|$	$\|T\|$ *(K)*	$\|V\|$ *(K)*	$\|S\|$	$\|T\|$ *(K)*	$\|V\|$ *(K)*
EMEA	***Fr/En***	1092.6	14.3/17.0	71.0/80.0	500	12.0/10.0	2.9/2.7	1000	27.0/21.0	4.5/4.5
	De/En	1108.8	13.3/14.5	128.0/71.0	500	10.0/10.0	3.2/2.8	1000	21.0/21.0	5.7/4.5
EU	***Es/En***	214.5	6.0/5.4	84.0/70.0	400	12.0/10.0	3.0/2.7	800	23.0/20.0	4.7/4.2
	Fr/En	982.7	20.7/18.9	161.4/150.4	400	11.5/10.1	2.9/2.6	800	22.5/20.0	4.5/4.0
TED	***Zh/En***	107	1.9/2.1	55.0/41.7	934	21.5/20.1	3.8/3.2	1664	33.2/31.9	4.5/3.7
	Es/En	160.2	3.0/3.2	89.0/61.7	887	19.1/20.1	4.1/3.4	1570	30.7/32.0	5.1/3.9
Xerox	***Es/En***	55.7	0.8/0.7	16.8/14.0	1012	16.0/14.4	1.8/1.6	1125	10.1/8.4	2.0/1.9
	Fr/En	51.8	0.5/0.6	24.8/13.7	964	10.7/10.9	1.7/1.5	984	11.9/12.5	2.2/1.8
Europarl	***Fr/En***	1991.2	60.5/54.5	160.0/131.2	3000	73.7/64.8	11.5/9.7	1500	29.9/27.2	6.3/5.6
	De/En	1902.2	49.8/52.3	394.6/129.1	3000	63.4/64.8	12.7/9.7	2169	44.1/46.8	10.0/8.1

[1] http://www.statmt.org/wmt14/medical-task/

[2] https://wit3.fbk.eu/mt.php?release=2013-01

[3] http://www.statmt.org/wmt15/translation-task.html

[4] The partition selected as development was *news-test2013*.

3.2 Metrics

The quality of our interactive protocol is assessed according to the following metrics:

Word Stroke Ratio (WSR) (Tomás and Casacuberta, 2006): Measures the number of words edited by the user, normalized by the number of words in the final translation. In this work, we assume that the edition of a word has a constant cost (one word stroke), independently of its length.

Mouse Action Ratio (MAR) (Barrachina et al., 2009): Measures the number of mouse actions made by the user, normalized by the number of characters in the final translation. In prefix-based IMT, the user makes a mouse action each time she needs to edit a word (to position the prompt), plus an additional action per sentence to validate the final translation. Segment-based IMT expands those mouse actions. Now, the user makes two actions each time she validates a segment (clicking at the beginning and at the end of the segment), and two more each time she deletes some words located between segments[5] (same procedure as selecting segments but using the right button of the mouse). In this work, we assume that the cost of a mouse action is more similar to the cost of typing a character than to the cost of typing a word. Therefore, we normalize the mouse actions with respect to characters.

Additionally, to evaluate the quality of the initial translations and the difficulty of each task, we used the following well-known metrics:

BiLingual Evaluation Understudy (BLEU) (Papineni et al., 2002): Computes the geometric average of the modified n-gram precision, multiplied by a brevity factor that penalizes short sentences.

Translation Error Rate (TER) (Snover et al., 2006): Computes the number of word edit operations (insertion, substitution, deletion and swapping), normalized by the number of words in the final translation.

3.3 Implementation

In this section, we present a simple implementation of a segment-based system using the XML scheme of the `Moses` toolkit. MT systems were trained with the standard configuration of `Moses`, using MERT (Och, 2003) for optimizing the weights of the log-linear model and estimating a 5-gram language model—using the improved KneserNey smoothing (Chen and Goodman, 1996)—with `SRILM` (Stolcke, 2002).

3.3.1 Prefix-Based Systems

Prefix-based IMT systems were implemented following the procedure described by Barrachina et al. (2009) of exploring a word graph and generating the best suffix for a given prefix. We generated a word graph for each sentence to translate. After that, treating the word graph as a weighted finite-state automaton, we parsed the validated prefix over the correspondent word graph—from the initial state to any other intermediate state—to find the best path that accounts for the prefix. Finally, we obtained the corresponding suffix searching for the best path from the intermediate state to the final state.

Our implementation of prefix-based IMT is, therefore, consistent with Barrachina et al. (2009), considering that we generate word graphs with the current version of the `Moses` toolkit.

3.3.2 Segment-Based Systems

Segment-based IMT systems were implemented following Domingo et al. (2016). Taking advantage of the `Moses` decoder XML markup scheme—which allows to specify the desired translation of parts of a sentence—we are able to validate segments of a translation hypothesis without altering the models. More precisely, we use the *exclusive* mode of this scheme, which only takes into account the given translation of a part of a sentence, ignoring any phrases from the

```
<x translation="If you have been exposed , you should " >Si vous avez été exposé , vous
devriez</x> consulter <x translation="your doctor for tests" >votre médecin pour des
tests</x>
```

Fig. 4 Example of a sentence in XML markup language (corresponding to the sentence of the first iteration of Fig. 2), specifying the desired translation for some parts of the sentence: *Si vous avez été exposé , vous devriez* must be translated as *If you have been exposed , you should* and *votre médecin pour des tests* as *your doctor for tests*.

phrase table that overlap with that span. With this, we can constrain the search process to follow Eq. 5. Fig. 4 shows an example of a sentence in XML markup language.

In order to manage the interaction between the user and the MT system, we built a prototype which takes into account the user feedback, generates a new translation with `Moses` and suggests the new hypothesis to the user. This has an average response time of 90 ms[6]—which, according to Nielsen (1993), is below *"the limit for having the user feel that the system is reacting instantaneously"*.

According to Section 2.2, the user feedback comes from three different actions: validating segments, correcting words and merging segments. The first two actions affect in the generation of the new XML markup sentence. Merging segments affects the system in the same way as validating segments. Therefore, we apply two different operations to the XML:

Segment validation: for each segment validated by the user, we align the words of that target segment with their correspondent source words (phrase alignments) and generate an XML tag to indicate the desired translation of those source words.

Word correction: each time the user corrects a word or inserts a new one, we align the new word with its correspondent source words using a hidden Markov alignment model (Vogel et al., 1996)—computing the alignment probability between the new word and the non-validated source words—and generate an XML tag to indicate that those source words have the validated word as a translation. These alignments are computed with GIZA++ (Och and Ney, 2003).

3.4 Confidence Measures

In order to profit from the word correction step of the process, we experiment with the use of confidence measures (CM). The aim is to correct first the word which leads to a largest improvement on the translation quality of the next hypothesis. To achieve this, the system suggests to the user which word she should correct first. We assume that correcting first the word with the least confidence leads to the largest improvement in future iterations. Therefore, the system suggests the non-validated word from the hypothesis with least confidence.

Following prior works that applied CM in IMT (Ueffing and Ney, 2005; González-Rubio et al., 2010), in this work we implement a word-level CM based on the IBM Model 1 (Brown et al., 1993), similar to the one described by Ueffing and Ney (2005). Additionally, we implement a word-level CM based on hidden Markov alignment models (Vogel et al., 1996). Given that time constraints are crucial in IMT, these implementations result suitable due to their speed. Given a source sentence $\mathbf{x} = x_1, \ldots, x_J$ and its translation hypothesis $\mathbf{y} = y_1, \ldots, y_I$, the confidence value of a word y_i ($c(y_i)$) is given by:

$$c(y_i) = \max_{0 \leq j \leq J} p(y_i \mid x_j) \tag{6}$$

where x_j is a source word at position j, J is the length of the source sentence, $p(y_i \mid x_j)$ is the lexicon probability given by either the IBM Model 1 or the hidden Markov alignment model and x_0 is the empty source word. Finally, we implement a random baseline, in which the word to correct is randomly selected.

[5] One mouse action is enough for selecting or deleting a one-word segment: the user would simply click on the word.

[6] Tested on a machine with an Intel i5 CPU at 3.1 GHz.

3.5 Evaluation on a Simulated Environment

Evaluation with human agents is too slow and expensive to be applied frequently during system deployment. For this reason, we carried out an automatic evaluation with simulated users whose desired translations are the reference sentences. User simulation is implemented accordingly to the different protocols to evaluate.

3.5.1 Prefix-Based Simulation

At each iteration, the user searches the leftmost wrong word from the translation hypothesis. Once that word is located, the user corrects it, validating a new prefix in the process. This correction has a cost of one mouse action and one word stroke. The system then reacts to this feedback, generating a new suffix that completes the prefix to conform a new translation hypothesis. This process is repeated until the hypothesis and the reference are the same.

3.5.2 Segment-Based Simulation

In this simulation, we assume that validated word segments must be in the same order as in the reference (the desired translation). For this reason, segments which should be reordered are not validated. Moreover, validated segments must maintain the same order in successive iterations. We are aware that more complex user models could contemplate the possibility of reordering validated segments. We left this as a future line of work. Additionally, when simulating the regular segment-based method, we assume, for the sake of simplicity and without loss of generality, that the user always corrects the leftmost wrong word. When simulating the segment-based method with the use of CM (see Section 3.4), we correct the word indicated by the system.

The simulation starts with the system producing an initial translation. Then, to simulate the segment validation, we compare this translation with the reference and compute the longest common subsequence (Apostolico and Guerra, 1987) between them. Once we obtain the common word segments, we validate them and increase the number of mouse actions—one action for each one-word segment, two actions for each multi-word segment. After that, to account for the word deletion, we check, from left to right, if any pair of consecutive validated segments should be merged into a single segment (i.e., they appear one after the other in the reference but are separated by some words in the hypothesis), in which case we delete the words between them (increasing mouse actions in one when deleting one word, and in two when deleting more than one word). Finally, to account for the word correction, in the regular segment-based method we compare translation and reference word by word. Once we find a difference, we input the reference word (increasing in one the number of mouse actions and word strokes). When using CM, the user always corrects the word indicated by the system. This is simulated by computing the confidence of each non-validated target word which is either next to a segment (the word right before or after the segment) or is the first or last word from the hypothesis. This limitation is necessary at the simulation in order to know the user correction. The corrected word is the one with the least confidence. Additionally, in the regular segment-based where the user is correcting from left to right, we inherently merge the corrected word with all the previous validated segments, creating a single validated segment. Finally, after the word correction, we generate the XML and obtain a new hypothesis. We repeat this process until the hypothesis matches the reference. Fig. 5 exemplifies this simulation.

3.6 XML Markup Scheme

The XML is constructed by associating the validated target segments with their corresponding source words. Analogously to the target side, we define a source segment as a word subsequence from the source sentence associated to a validated segment. In this section, we show and discuss some problems arisen in the implementation of the segment-based protocol and the design decisions taken for overcoming them.

Reference: If you have been exposed , you should go to your doctor for tests
Hypothesis: If you have been exposed , you should consult go your doctor for tests

Segment validation: [If you have been exposed , you should] consult [go] [your doctor for tests]
Mouse actions: $2 + 1 + 2 = 5$

Words deletion: [If you have been exposed , you should ~~consult~~ go] [your doctor for tests]
Mouse actions: 1

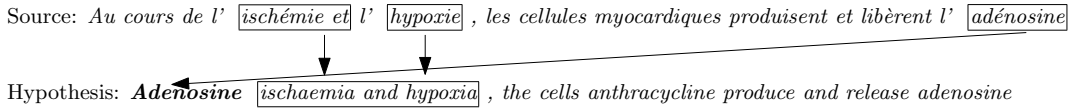
Word correction: [If you have been exposed , you should go] **to** [your doctor for tests]
Mouse actions: 1
Word strokes: 1

Total mouse actions: 7
Total word strokes: 1

Fig. 5 Follow up to the example in Fig 3 to exemplify how user actions are simulated. In the **segment validation**, we compute the longest common subsequence between hypothesis and reference, obtaining the segments: *If you have been exposed , you should*, *go* and *your doctor for tests*). After that, in the **word deletion**, since the first two validated segments appear together in the reference, we delete the word between them (*consult*) to create a bigger validated segment (*If you have been exposed , you should go*). Finally, in the **word correction**, we look for the leftmost reference word not included in a validated segment (to) and add it to the target in its correspondent position. Validating or deleting words have a cost of one mouse action for one-word segments, and two mouse actions for multiple-word segments. A word correction has a cost of one mouse action and one word stroke.

3.6.1 Non-Consecutive Corresponding Sources

A validated segment might be aligned with more than one source segment. In those cases if, when generating the XML, we assign to each source segment their correspondent translation, we might end up altering the order of the words in the target segment. To avoid this, we assign the complete target segment as the desired translation of the leftmost source segment, and assign an empty translations to the rest of the source segments. Fig. 6 shows an example in which this situation happens.



XML: `Au cours de l' <x translation="ischaemia and hypoxia" >ischémie et</x> l' <x translation=" " >hypoxie</x> , les cellules myocardiques produisent et libèrent l' <x translation="Adenosine" >adénosine</x>`
Translation: *During* [*Adenosine*] *the cells , produce and release the myocardial* [*ischaemia and hypoxia*]

Fig. 6 Example of a sentence in XML markup language in which a validated segment (*ischaemia and hypoxia*) has been originated by more than one source segment (*ischémie et*, *hypoxie*). The leftmost source segment (*ischémie et*) is assigned the translation (*ischaemia and hypoxia*), and the rest (*hypoxie*) is assigned an empty translation (" ").

3.6.2 Segment Reorders

According to the user model (see Section 3.5.2), the user validates segments taking into account their order of appearance in the hypothesis. For this reason, we need to keep the segment ordering from one iteration to the next one. To achieve this,

we make use of the *wall* reordering constraint, which ensures that all words left to a wall are translated before translating the rest of the sentence. This limits the reordering model, making it unable to reorder words located in different sides of a wall. Fig. 7 shows and example of a sentence in XML using walls.

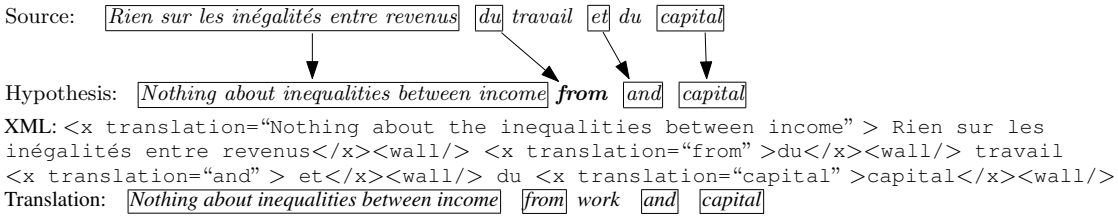


Fig. 7 Example of a sentence in XML markup language using the wall reordering constraint (corresponding to the same example as Fig. 4).

An additional reordering problem appears if source and target validated segments are ordered differently. This can cause a wrong reordering of the target segments in the successive translation hypotheses. Since our user model assumes that validated segments will not be reordered, we must ensure that the ordering is kept along the process. To achieve this, after generating the translation with `Moses`, we reorder the validated segments to match the ordering provided by the user. Fig. 8 shows and example of constructing a translation using this solution.

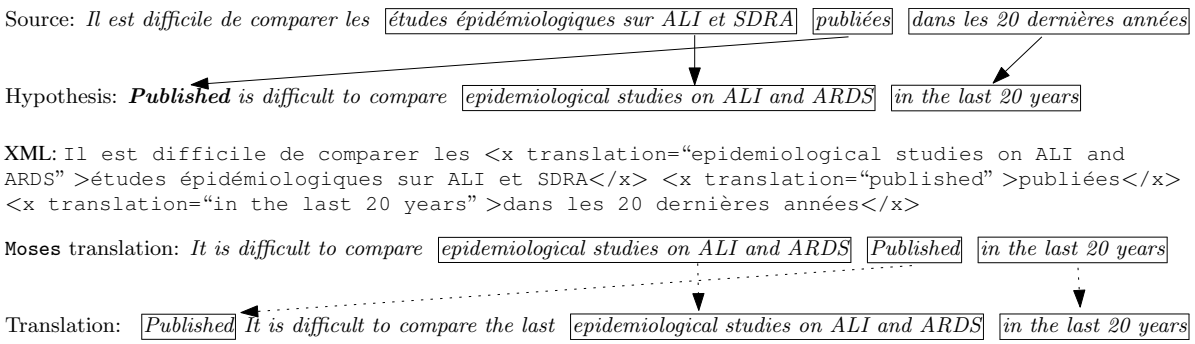


Fig. 8 Example of a sentence in XML markup language in which source and target are ordered differently. The user has validated three segments (**Published** , epidemiological studies on ALI and ARDS and in the last 20 years). However, due to the difference in order between source and target, these segments are reordered in the new hypothesis (epidemiological studies on ALI and ARDS , **Published** and in the last 20 years). As a solution, after creating the XML and generating the translation with `Moses`, we reorder the translation to ensure the order indicated by the user. Arrows represent alignments between source and target validated segments. Dashed arrows represent the change in position of target validated segments.

Since we construct the translation following the source segment order, this solution affects the language model. This is due to the XML scheme being strongly affected by the order in which the translation is constructed, and results in the need of reordering the translation generated by `Moses`. An alternative solution is to modify the way in which we construct the XML. Instead of assigning to each source segment its correspondent target segment (as in the previous strategy), we match source and target segments consecutively: the first source segment with the first target segment; the second source segment with the second target segment; etc. Fig. 9 shows an example of this solution.

With this strategy, the language model is unaffected. However, the translation assigned to a given source segment might not be the real one. We tested both approaches, observing that penalizing the language model is more severe than affecting the translation and reordering models. Therefore, we followed the second strategy.

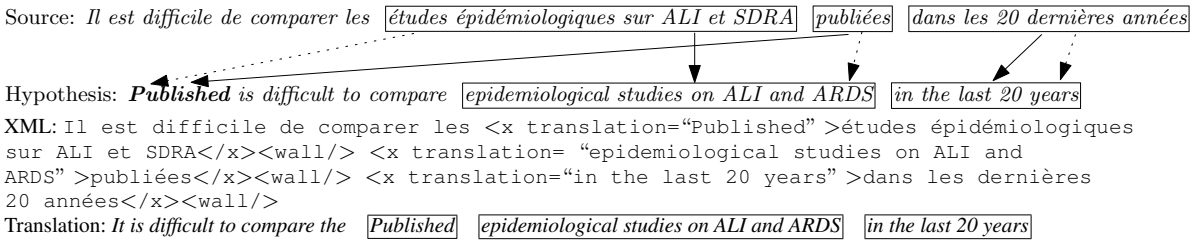


Fig. 9 Alternative to the solution in Fig. 8 of a sentence in XML markup language in which source and target are ordered differently. Dashed arrows represent how these segments have been aligned in the XML. To ensure that, in the new translation, the validated segments respect the order indicated by the user, we modify the way in which we construct the XML. Now, instead of their corresponding translation, the first source segment (*études épidémiologiques sur ALI et SDRA*) is assigned the translation of the first target segment (*Published*), and the second source segment (*publiées*) is assigned the translation of the second target segment (*epidemiological studies on ALI and ARDS in the last 20 years*). Arrows represent alignments between source and target validated segments.

3.6.3 Words without Corresponding Sources

Each time the user makes a word correction, we need to find its corresponding source words to generate the XML (see Section 3.3.2). However, if the new word is an out-of-vocabulary or its alignment probability is very low, we are unable to find them and, therefore, cannot update the XML to account for the word correction. To solve this problem, we artificially add a new source at the end of the segment, and generate the XML considering this artificial source as the corresponding source of the word corrected by the user. Fig. 10 shows an example in which this situation appears.

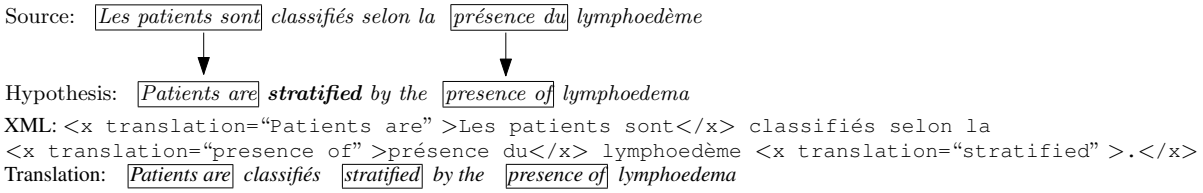


Fig. 10 Example of a sentence in XML markup language in which we were unable to find the corresponding source words of the user word correction (***stratified***), due to the low probability of aligning it with its original source word (*classifiés*). As a solution, we artificially add a new source (.) at the end of the sentence and assign the word correction as its translation.

3.6.4 Spurious Words

The spurious words represent a challenging problem for our system. We refer to source words which do not have a direct correspondence with the words from the desired translation. Since the XML is generated by assigning the validated targets to their correspondent sources, spurious words never get into the XML and are always translated by `Moses`, generating undesired translations. Additionally, in those cases in which we cannot identify the sources of a user word correction (see Section 3.6.3), a similar problem appears: `Moses` generates new translations for those source words, because their translation was not specified.

These untreated sources generate undesired translations, resulting in an increase of the user effort who, in order to obtain the desired translation, needs to merge more segments or to increase the number of times that she inputs an end-of-translation stroke. This problem represents a major challenge within our proposal. We aim to address it in future works. Fig. 11 and Fig. 12 reflect this problem.

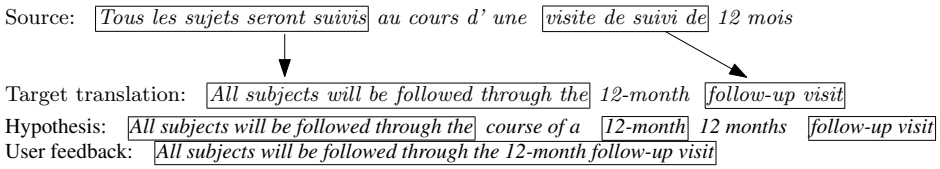


Fig. 11 Example of the increase in the number of mouse actions due to spurious source words. The source words *au cours d' une* do not have a target translation, but `Moses` translates them as *course of a*. Additionally, at some point of the session, the user correction *12-month* has failed to identify their correspondent source words, and so `Moses` is generating an undesired translation for them (*12 months*) For this reason, prior to validating the translation, the user has to perform two additional merge operations.

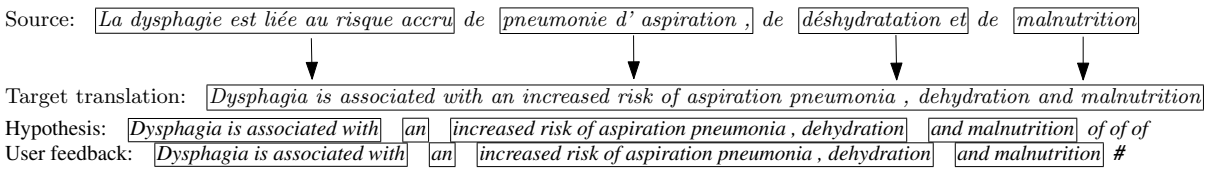


Fig. 12 Example of the increase in the number of word strokes due to spurious source words. The source words *de*, *de* and *de* do not have a target translation, but `Moses` translates them as *of of of*. Prior to validating the translation, the user must type the special end-of-translation stroke (#) to indicate to the system that the validated parts of the hypothesis conform her desired translation.

4 Results

In this section, we present and discuss the experimental results obtained. First, we compare the segment-based against the prefix-based approach. After that, we discuss the results of using CM. Finally, we qualitative analyze the main weaknesses of our approach.

4.1 Quantitative Analysis

Table 2 compares the user effort results of the segment-based against the prefix-based approach. Prefix-based results were obtained following Barrachina et al. (2009) and are similar to those reported in the literature (Tomás and Casacuberta, 2006; Barrachina et al., 2009), taking into account that we are generating the word graphs using `Moses` version 3. The quality of the initial translation is shown as indicative of the difficulty of each task.

The segment-based approach clearly improves the prefix-based in terms of the effort required for typing corrections (yielding diminishes of up to 47 points of WSR). However, this reduction comes with an increase in the number of mouse actions (from 5 up to 25 points of MAR), which is always smaller than the effort reduction.

In the case of the EMEA corpus, the segment-based approach obtains a reduction of 17 to 40 points of WSR, at the expenses of increasing the MAR in 10 points. Since the initial translation quality of the French–English tasks is higher than the German–English tasks, this last pair of languages obtains the highest effort reduction.

Something similar happens with the EU corpus. In this case, the initial translation quality is higher for all language pairs. Therefore, the effort reduction is smaller. Nonetheless, the segment-based approach obtains a typing reduction of 10 to 15 points of WSR, with an increase in the number of mouse actions of 5 to 6 points of MAR.

The TED corpus performs the highest effort reduction, since it contains the language pair with the lowest initial quality translation (Chinese–English, with 8.7/11.7 points of BLEU and 83.3/76.2 points of TER). In this case, the effort reduction consists in an improvement of 26 to 47 points of WSR, at the expenses of an increase of 13 to 25 points of MAR. It is worth mentioning the high value of the mouse effort when translating to Chinese, which is most likely due to this language containing very few characters per word. The Spanish–English tasks, containing a higher initial translation

Table 2 Results of the segment-based IMT approach in comparison with the prefix-based approach. All values are reported as percentages.

Corpus	Language	BLEU	TER	Prefix-Based		Segment-Based	
				WSR	MAR	WSR	MAR
EMEA	Fr–En	30.5	48.6	57.8	12.4	33.6	21.6
	En–Fr	29.8	52.6	58.4	12.5	41.7	21.7
	De–En	23.4	57.6	70.9	14.1	31.0	24.4
	En–De	15.7	64.8	74.9	12.0	35.6	23.1
EU	Es–En	47.3	40.8	45.6	10.2	30.5	16.0
	En–Es	47.9	41.1	44.6	9.7	31.9	14.8
	Fr–En	52.1	36.2	37.3	7.5	26.3	14.4
	En–Fr	51.3	38.6	38.8	7.3	29.4	12.8
TED	Zh–En	11.7	76.2	83.1	22.4	36.1	35.8
	En–Zh	8.7	83.3	86.3	55.7	60.0	80.0
	Es–En	36.5	42.7	51.1	12.9	31.7	22.9
	En–Es	31.3	47.7	53.2	12.3	36.7	22.8
Xerox	Es–En	52.2	31.8	35.8	10.5	20.0	20.4
	En–Es	60.8	27.3	28.3	7.9	21.9	14.3
	De–En	32.2	54.6	62.7	15.1	29.2	26.9
	En–De	24.1	64.5	68.3	12.6	32.7	23.6
Europarl	Fr–En	26.5	51.4	58.7	13.9	30.2	30.3
	En–Fr	26.5	55.6	61.4	13.5	31.5	28.4
	De–En	19.2	61.1	73.3	17.7	34.4	30.8
	En–De	15.3	68.4	75.0	15.0	33.1	25.9

quality, are more similar to the previous task. They obtain a 20 points reduction of the typing effort, with an increase of 10 points of the mouse effort.

The Xerox corpus has similar results to the previous corpora. The Spanish–English tasks contain a higher initial translation quality, and so the effort reduction is lower (7 to 15 points of WSR at the expenses of an increase of 6 to 10 points of MAR). The German–English tasks, having a lower translation quality, have 33 to 36 points of reduction of the typing effort, and an increase of 11 points of the mouse effort.

In the case of the Europarl corpora, both language pairs behave similarly, obtaining a typing effort reduction of 28 to 35 points of WSR, with an increase in the number of mouse actions of 11 to 16 points of MAR.

Finally, the experiments in which we used CM to select which word to correct first (see Section 3.4) have been unsuccessful. Correcting first the non-validated word with the lowest confidence value has failed at improving the translation quality of the next hypothesis, resulting in the same amount of user effort (both in terms of word corrections and mouse actions). Table 3 shows the results comparing the regular segment-based approach (which always corrects the leftmost wrong word first) with the world-level CM approaches based on IBM model 1 and hidden Markov alignment models, and the random baseline. All strategies obtained similar results, which leads to the conclusion that the order in which corrections are made does not affect the overall user effort.

4.2 Qualitative Analysis

To better understand the experimental results, we display some examples which reflect the system's weaknesses.

Table 3 Results of the segment-based approach using CM to select the order in which words are corrected. In the regular segment-based, the word corrected its the leftmost wrong word. IBM_1 implements world-level CM based on IBM model 1. HMM implement world-level CM based on hidden Markov alignment models. Random is a baseline in which the word to correct is selected randomly. All values are reported as percentages.

Corpus	Language	Segment-Based		CM					
		WSR	MAR	IBM_1		HMM		Random	
				WSR	MAR	WSR	MAR	WSR	MAR
EMEA	Fr–En	33.6	21.6	35.1	23.4	35.5	22.9	35.7	22.8
	En–Fr	41.7	21.7	41.2	23.3	41.8	22.5	41.9	22.0
	De–En	31.0	24.4	30.3	24.3	30.7	24.6	30.0	24.1
	En–De	35.6	23.1	35.0	22.6	35.2	22.6	34.7	22.6
EU	Es–En	30.5	16.0	30.7	17.6	31.2	17.2	31.0	17.0
	En–Es	31.9	14.8	31.2	16.7	31.6	16.0	31.7	15.8
	Fr–En	26.3	14.4	26.9	15.7	27.2	15.5	27.2	15.4
	En–Fr	29.4	12.8	29.4	13.8	29.6	13.7	29.6	13.5
TED	Zh–En	36.1	35.8	35.8	35.4	35.9	35.4	34.9	35.0
	En–Zh	60.0	80.0	60.3	85.5	60.9	83.3	60.9	81.8
	Es–En	31.7	22.9	32.0	24.7	32.3	24.4	32.2	24.2
	En–Es	36.7	22.8	36.6	24.7	37.1	24.0	37.1	23.7
Xerox	Es–En	20.0	20.4	20.1	20.4	20.1	20.5	19.9	20.1
	En–Es	21.9	14.3	22.3	15.2	22.6	14.9	22.6	14.7
	De–En	29.2	26.9	29.3	26.7	29.2	26.6	29.0	26.5
	En–De	32.7	23.6	32.1	22.6	32.3	22.5	32.0	22.7
Europarl	Fr–En	30.2	30.3	29.8	29.7	29.8	29.7	29.4	29.6
	En–Fr	31.5	28.4	30.9	27.7	31.1	27.6	30.4	27.5
	De–En	34.4	30.8	34.3	30.7	34.5	30.7	33.6	30.2
	En–De	33.1	25.9	32.6	25.4	32.6	25.4	32.1	25.3

Fig. 13 represents an example in which the spurious words problem (see Section 3.6.4) is noteworthy. The session starts with the system proposing an initial translation. Then, the user validates some word segments and makes a correction. However, this correction (***Early-onset***) is an out-of-vocabulary word and thus, the system is unable to associate it with their correspondent sources (*apparition précoce*) and keeps offering a translation for them in following iterations. Therefore, at the next iteration, besides making a correction (there are no new correct word segments to validate), the user has to do a merge operation (uniting the first validated segment with the start of the sentence) to delete those undesired translated words. This process continues in a similar way during the rest of the iterations, with the user having to merge more segments to cope with the problem. The correction made at iteration two (***occurring***) produces also an error, increasing the problem further. This, together with the spurious words contained in the source sentence (*L'*, *de*, *la* and *septicémie*), results in the user having to make ten extra mouse actions (two per each pair of segments merged) to cope with the problem.

The problem of having translations of spurious words and words for which the user has already typed a translation is fairly common. Although, in many cases, it only consists in a few words at some point of the session and does not have a cumbersome effect, this problem is present in more than half of the cases.

Finally, Fig. 14 depicts a case in which the system has an undesired behavior. Due to the combination of containing an out-of-vocabulary word (*gens*), 4 spurious words (*un*; ,; *la* and *l'*) and a noteworthy word reorder (the first and second

source (x): L' apparition précoce de la septicémie néonatale est définie comme une septicémie qui se produit dans les 7 premiers jours de vie

target translation ($\hat{y}$): Early-onset neonatal sepsis is defined as occurring within the first 7 days of life

IT-0	**MT**	The onset early neonatal sepsis is defined as sepsis which occurs within 7 days of life
IT-1	**User**	**Early-onset** onset early [neonatal sepsis is defined as] sepsis which occurs [within] [7 days of life]
	MT	*The onset of the early* [Early-onset] [neonatal sepsis is defined as] sepsis which occurs [within] [7 days of life]
IT-2	**User**	[Early-onset] [neonatal sepsis is defined as] **occurring** which occurs [within] [7 days of life]
	MT	[Early-onset] *early development of* [neonatal sepsis is defined as] *sepsis which occurs* [occurring] [within] [7 days of life]
IT-3	**User**	[Early-onset neonatal sepsis is defined as occurring] [within] **the** [7 days of life]
	MT	[Early-onset neonatal sepsis is defined as occurring] [within] *early development* [the] *sepsis which product* [7 days of life]
IT-4	**User**	[Early-onset neonatal sepsis is defined as occurring] [within the] **first** *which product* [7 days of life]
	MT	[Early-onset neonatal sepsis is defined as occurring] [within the] *early onset sepsis which product* **first** [7 days of life]
IT-5	**User**	[Early-onset neonatal sepsis is defined as occurring] [within the first] [7 days of life]
	MT	[Early-onset neonatal sepsis is defined as occurring] [within the first] [7 days of life]
END	**User**	Early-onset neonatal sepsis is defined as occurring within the first 7 days of life

Fig. 13 Example of a segment-based IMT session in which the spurious words problem results in a cumbersome behavior. Words in *italic* represent undesired translations produced by the system.

halves of the source sentence are reordered in the target sentence), the system fails at constructing good translations. In fact, the initial hypothesis only contains two correct word segments of one word each. As a result, the user not only has to type more word corrections but she also has to merge more segments. In this case, however, most of the increases in merge operations are not due to the spurious word problem but to the system failing in reordering the translation. The untranslated part of the first half of the source sentence keeps getting translated after the first validated segments, and so the user has to merge segments to delete those undesired translated words.

Manually post-editing the initial hypothesis would have taken 10 word strokes plus 11 mouse actions, and the segment-based approach has taken 8 word strokes plus 33 mouse actions. Nonetheless, this is an infrequent example of the system's behavior.

4.3 Discussion

The segment-based approach succeeds at reducing the user typing effort, taking advantage of the correct parts of each translation hypothesis. In general, tasks with the lowest translation quality are the ones with the greatest effort reduction. However, since the user is validating those correct parts, the mouse effort increases. Nonetheless, this increase is smaller than the typing reduction and, thus, it pays off.

An exception to this is the English–Chinese task of the TED corpus. In this task, the system is unable to take profit of the user's corrections, and successive hypothesis does not improve in translation quality. This results in a smaller reduction of the typing effort. Moreover, the mouse effort is greatly increased. Therefore, in this case, the typing effort reduction and the mouse effort increase are similar and, thus, the user effort does not improve.

Furthermore, as seen in Section 4.2, there are cases in which the system has an undesired behavior. Both, source spurious words and user corrections which produce an error, result in an increase in the number of mouse actions. The user has to merge more segments in order to delete those undesired translations, which can become cumbersome. However, this undesired behavior happens infrequently. Generally, in those cases in which these problems are present, it only consists in a few words at some point of the sessions and does not have this cumbersome effect. Therefore, considering the typing effort reduction, the user effort improves despite this increase in the number of mouse actions. Nonetheless, we

source ($\mathbf{x}$): À un certain moment de leur vie , la plupart des gens vont souffrir de l' acné
target translation ($\hat{\mathbf{y}}$): Most people will suffer from acne at some point in their life

IT-0	**MT**	To a certain time of life , most gens will experience acne
IT-1	**User**	**Most** a certain time of life , most gens will experience acne
	MT	To a certain time of life , the Most of gens will experience acne
IT-2	**User**	Most **people** gens will experience acne
	MT	Most To a certain time of life , *the of* people will experience acne
IT-3	**User**	Most people will **suffer** acne
	MT	Most people will To a certain time of life , *the of* suffer acne
IT-4	**User**	Most people will suffer **from** acne
	MT	Most people will suffer To a certain time from their life *the* acne
IT-5	**User**	Most people will suffer from acne **at**
	MT	Most people will suffer from acne *To a* certain at their life *the*
IT-6	**User**	Most people will suffer from acne at **some** their life *the*
	MT	Most people will suffer from acne at *To a* some their life *, the*
IT-7	**User**	Most people will suffer from acne at some **point** their life *, the*
	MT	Most people will suffer from acne at some *To a* point *the European* their life
IT-8	**User**	Most people will suffer from acne at some point **in** *European* their life
	MT	Most people will suffer from acne at some point *To a* in , their life
IT-9	**User**	Most people will suffer from acne at some point in their life
	MT	Most people will suffer from acne at some point in their life
END	**User**	Most people will suffer from acne at some point in their life

Fig. 14 Example of a segment-based IMT session in which the system has an undesired behavior. Words in *italic* represent undesired translations produced by the system.

should test our proposal with real users to better measure this effort reduction. Additionally, as a solution to this problem, we want to explore the use of confidence measures to detect those source words which should not have a translation.

Finally, using CM to assist the user in the correction step has failed at improving the effort reduction. We have tested different strategies, resulting in the same overall user effort. Due to the way in which the XML scheme works, the word corrected by the user only affects those phrases located near that word. Therefore, altering the order in which words are corrected changes which parts of the sentence are corrected first but, overall, results in the same user effort.

5 Conclusions and Future Work

In this work, we have formally presented an IMT protocol that allows the user to validate the correct parts of a translation hypothesis. We have carried out a simple, but effective, implementation of the protocol using a feature of the `Moses` toolkit. Tested in a simulated environment, we have compared this segment-based approach against the classical prefix-based protocol. Results show that the segment-based approach succeeds in overcoming the prefix-based limitation of only correcting the prefix, resulting in a reduction of the user effort. This effort improvement results in a substantial decrease of the typing effort, at the expenses of an increase in the number of mouse actions.

Part of this increase of the mouse effort is due to the system failing to find the corresponding sources of the user word corrections. These sources generate undesired translations, resulting in the user having to merge more segments to

cope with this problem. Additionally, spurious words from the source sentence result in a similar problem: the system translates them and the user has to do more mouse actions to deal with them.

The segment-based methodology successfully takes advantage of the correct parts of a translation hypothesis. This is reflected in the results of the tasks which had the lowest initial translation quality. With one exception, these tasks have been the ones to have the greatest improvement of the user effort. This exception has been the English–Chinese task which, unable to take advantage from them, has needed a greater number of user corrections.

We have also tested an active interaction protocol to assist the user in the correction step of the process. In this protocol, the system informed the user about which word should be corrected first to improve the quality of the next hypothesis. We implemented this protocol using different approaches, relying on the use of confidence measures. We obtained similar results with each of them. Therefore, we concluded that changing the order in which words are corrected had no effect in the overall user effort. The XML scheme takes profit from the word correction only to generate those phrases located near that word. Therefore, the only effect that altering the order in which the user makes corrections has is to change which parts of the sentence are corrected first.

As future work, we need to improve the way in which the system finds the corresponding source words of a user correction, and the problem with spurious words contained in the source sentence. Additionally, we want to develop new protocols to assist the user in the segment validation step of the process. Furthermore, our user model only validated segments which were ordered in the same way as in the desired translation. In future works, we want to explore other approaches—such as allowing the user to reorder segments. Finally, in this work we assume that making a mouse action is less of an effort than typing a word and, thus, that the increase in the mouse effort pays off with respect to the significant reduction of the typing effort. However, we should test our proposal with real users to obtain actual measures of the effort reduction.

Acknowledgements The research leading to these results has received funding from the Ministerio de Economía y Competitividad (MINECO) under project CoMUN-HaT (grant agreement TIN2015-70924-C2-1-R), and Generalitat Valenciana under project ALMAMATER (grant agreement PROMETEOII/2014/030).

References

Alabau, V., Bonk, R., Buck, C., Carl, M., Casacuberta, F., García-Martínez, M., González-Rubio, J., Koehn, P., Leiva, L. A., Mesa-Lao, B., Ortiz-Martínez, D., Saint-Amand, H., Sanchis-Trilles, G., and Tsoukala, C. (2013). CASMACAT: An open source workbench for advanced computer aided translation. *The Prague Bulletin of Mathematical Linguistics*, 100:101–112.

Alabau, V., Rodríguez-Ruiz, L., Sanchis, A., Martínez-Gómez, P., and Casacuberta, F. (2011). On multimodal interactive machine translation using speech recognition. In *Proceedings of the International Conference on Multimodal Interaction*, pages 129–136.

Alabau, V., Sanchis, A., and Casacuberta, F. (2014). Improving on-line handwritten recognition in interactive machine translation. *Pattern Recognition*, 47(3):1217–1228.

Apostolico, A. and Guerra, C. (1987). The longest common subsequence problem revisited. *Algorithmica*, 2:315–336.

Azadi, F. and Khadivi, S. (2015). Improved search strategy for interactive machine translation in computer-asisted translation. In *Proceedings of Machine Translation Summit XV*, pages 319–332.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (arXiv:1409.0473)*.

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

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pages 310–318.

Cheng, S., Huang, S., Chen, H., Dai, X., and Chen, J. (2016). Primt: A pick-revise framework for interactive machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 1240–1249.

Dale, R. (2016). How to make money in the translation business. *Natural Language Engineering*, 22(2):321–325.

Domingo, M., Peris, Á., and Casacuberta, F. (2016). Interactive-predictive translation based on multiple word-segments. In *Proceedings of the Annual Conference of the European Association for Machine Translation*, pages 282–291.

Federico, M., Bentivogli, L., Paul, M., and Stüker, S. (2011). Overview of the IWSLT 2011 evaluation campaign. In *International Workshop on Spoken Language Translation*, pages 11–27.

Foster, G., Isabelle, P., and Plamondon, P. (1997). Target-text mediated interactive machine translation. *Machine Translation*, 12:175–194.

González-Rubio, J., Benedí, J.-M., and Casacuberta, F. (2016). Beyond prefix-based interactive translation prediction. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning*, pages 198–207.

González-Rubio, J., Ortiz-Martínez, D., and Casacuberta, F. (2010). On the use of confidence measures within an interactive-predictive machine translation system. In *Proceedings of the Annual Conference of the European Association for Machine Translation*.

Knowles, R. and Koehn, P. (2016). Neural interactive translation prediction. In *Proceedings of the Association for Machine Translation in the Americas*, pages 107–120.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Machine Translation Summit*, pages 79–86.

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 177–180.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54.

Koehn, P., Tsoukala, C., and Saint-Amand, H. (2014). Refinements to interactive translation prediction based on search graphs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 574–578.

Marie, B. and Max, A. (2015). Touch-based pre-post-editing of machine translation output. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1040–1045.

Nepveu, L., Lapalme, G., Langlais, P., and Foster, G. (2004). Adaptive language and translation models for interactive machine translation. In *Proceedings of the Conference on Empirical Method in Natural Language Processing*, pages 190–197.

Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann Publishers Inc.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

Och, F. J. and Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 295–302.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistic*, 29(1):19–51.

Ortiz-Martínez, D. (2016). Online learning for statistical machine translation. *Computational Linguistics*, 42(1):121–161.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Peris, Á., Domingo, M., and Casacuberta, F. (2017). Interactive neural machine translation. *Computer Speech & Language*, 45:201–220.

Sanchis-Trilles, G., Ortiz-Martínez, D., Civera, J., Casacuberta, F., Vidal, E., and Hoang, H. (2008). Improving interactive machine translation via mouse actions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 485–494.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas*, pages 223–231.

Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 257–286.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *NIPS*, volume 27, pages 3104–3112.

Tiedemann, J. (2009). News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*, volume V, pages 237–248.

Tomás, J. and Casacuberta, F. (2006). Statistical phrase-based models for interactive computer-assisted translation. In *Proceedings of the International Conference on Computational Linguistics/Association for Computational Linguistics*, pages 835–841.

Torregrosa, D., Forcada, M. L., and Pérez-Ortiz, J. A. (2014). An open-source web-based tool for resource-agnostic interactive translation prediction. *Prague Bulletin of Mathematical Linguistics*, 102:69–80.

Tseng, H., Chang, P., Andrew, G., Jurafsky, D., and Manning, C. (2005). A conditional random field word segmenter. In *Proceedings of the Special Interest Group of the Association for Computational Linguistics Workshop on Chinese Language Processing*, pages 168–171.

Ueffing, N. and Ney, H. (2005). Application of word-level confidence measures in interactive statistical machine translation. In *Proceedings of the European Association for Machine Translation*, pages 262–270.

Vogel, S., Ney, H., and Tillmann, C. (1996). HMM-based word alignment in statistical translation. In *Proceedings of the Conference on Computational Linguistics*, volume 2, pages 836–841.

Wuebker, J., Green, S., DeNero, J., Hasan, S., and Luong, M.-T. (2016). Models and inference for prefix-constrained machine translation. In *Proceedings of the Annual Meeting of the Association for the Computational Linguistics*, pages 66–75.

Zens, R., Och, F. J., and Ney, H. (2002). Phrase-based statistical machine translation. In *Proceedings of the Annual German Conference on Advances in Artificial Intelligence*, volume 2479, pages 18–32.