Online Learning for Effort Reduction in Interactive Neural Machine Translation

Álvaro Peris*, Francisco Casacuberta

Pattern Recognition and Human Language Technology Research Center,
Universitat Politècnica de València,
Camino de Vera s/n, 46022 Valencia, SPAIN

Abstract

Neural machine translation systems require large amounts of training data and resources. Even with this, the quality of the translations may be insufficient for some users or domains. In such cases, the output of the system must be revised by a human agent. This can be done in a post-editing stage or following an interactive machine translation protocol.

We explore the incremental update of neural machine translation systems during the post-editing or interactive translation processes. Such modifications aim to incorporate the new knowledge, from the edited sentences, into the translation system. Updates to the model are performed on-the-fly, as sentences are corrected, via online learning techniques. In addition, we implement a novel interactive, adaptive system, able to react to single-character interactions. This system greatly reduces the human effort required for obtaining high-quality translations.

In order to stress our proposals, we conduct exhaustive experiments varying the amount and type of data available for training. Results show that online learning effectively achieves the objective of reducing the human effort required during the post-editing or the interactive machine translation stages. Moreover, these adaptive systems also perform well in scenarios with scarce resources. We show that a neural machine translation system can be rapidly adapted to a specific domain, exclusively by means of online learning techniques.

Keywords: Neural machine translation; Interactive machine translation; Machine translation post-editing; Online learning; Domain adaptation; Deep learning.

1. Introduction

In the last years, the field of machine translation (MT) has witnessed impressive progresses, mostly due to the advances achieved in corpus-based machine translation. Nowadays, MT systems are useful tools for many users and companies, automatically providing translations of acceptable quality in many cases (Crego et al., 2016; Wu et al., 2016).

Nevertheless, we are still far from solving the MT problem (Koehn and Knowles, 2017). The systems still produce wrong translations, that may be intolerable for some users or domains. For example, translations of medical records must be accurate and error-free. Moreover, the translation problem has many subtleties that make it hard for machines to tackle it: discourse adequacy, anaphora resolution, domain-specific meanings, stylistic forms, etc.

In scenarios that require high-quality translations, the outputs of the systems are usually reviewed by a human agent, who corrects the errors made by the MT system. Thus, the user is benefited by the use of

*Corresponding author: Phone: (34) 96 387 70 69
Email addresses: lvapeab@prhlt.upv.es (Álvaro Peris), fcn@prhlt.upv.es (Francisco Casacuberta)

MT, since it allows to achieve higher productivity than translating from scratch (Arenas, 2008; Green et al., 2013a). The process of correcting MT hypotheses is known as post-editing.

Given the advantages of post-editing, new approaches to efficiently produce good translations emerged, aiming to increase the productivity of this process, thereby diminishing the human effort required. Among them, interactive machine translation (IMT) (Barrachina et al., 2009; Casacuberta et al., 2009; Foster et al., 1997) is one of the most attractive strategies.

The IMT framework introduces the human corrector into the editing process: the system reacts to each human action. As the user makes a correction, the system has more information for generating an alternative translation, hopefully better than the previous one, which spares effort to the human translator.

With translation post-editing or IMT, we obtain high-quality translations. These translations contain new knowledge, which is prone to be profited by an adaptive MT system, taking advantage of these new samples and adapting its models to them. Online learning (OL) methods are suitable for this goal. OL is a machine learning paradigm in which data is available sequentially and models are updated incrementally, sample to sample. The online learning framework can be structured according to four main stages (Murphy, 2012):

1. A sample is presented to the system.
2. The system provides a prediction for this sample.
3. The correct prediction is provided to the system.
4. The system uses the correct prediction to adapt its models.

Therefore, in the OL framework, there is no distinction between the training and prediction phases: the system is continuously learning and predicting, as data become available. Therefore, the MT systems can be retrained as the correction process goes on, avoiding to make the same errors again and being adapted to a given domain or tailored to the style of the human corrector.

The MT technology has recently experienced a revolution. During several years, phrase-based statistical machine translation (PB-SMT) models (Koehn et al., 2003) were the state-of-the-art in MT. But in the last years, a novel corpus-based technique emerged: the so-called neural machine translation (NMT), in which the translations are generated solely by neural networks. NMT achieves more fluent and natural translations (Wu et al., 2016) than previous PB-SMT systems. Moreover, NMT systems perform exceptionally well under an IMT paradigm, as shown by Knowles and Koehn (2016) and Peris et al. (2017b).

But the NMT technology also has weaknesses. As studied by Koehn and Knowles (2017), NMT systems have lower quality when translating out-of-domain sentences and they require larger amounts of training data than PB systems. Given these findings, NMT faces a dilemma: on the one hand, training with large amounts of in-domain data may be infeasible, due economic restrictions or to the lack of domain-specific data. On the other hand, we need to train with large amounts of data, in order to obtain good translations. A possible solution for this issue relies on the domain adaptation field (Ben-David et al., 2010). In this scenario, a model trained on a large corpus of out-of-domain samples, aims to perform well on a different domain. In the field of MT, under the domain adaptation umbrella, we find a large set of techniques (Axelrod et al., 2011; Chen et al., 2017; Farajian et al., 2017).

In this work, we aim to leverage the aforementioned issues of NMT systems. Our goal is to adapt translation systems on-the-fly, during the post-editing or IMT stages. For doing this, we take advantage of the human-revised translations generated in the post-editing or IMT processes and apply OL techniques to the NMT system. Additionally, we deepen into the application of OL into IMT framework, using the NMT technology. To the best of our knowledge, this is the first work that puts together interactive neural machine translation and online learning. We thoroughly evaluate our models, setting up three different scenarios, that account for the casuistic that can happen in an industrial MT setting. We show that OL can be effectively used for enhancing the NMT system and reducing the human effort required during the correction process. Our main contributions are:

1. We study the application of OL techniques in the post-editing and IMT scenarios, using NMT systems.
2. We introduce a simple and effective way for performing character-level interactions in a (sub)word-based interactive NMT (INMT) system.

3. We conduct a large experimentation, using public corpora containing different features from varied domains. We stress the translation systems, applying them in three different translation scenarios.
4. We show that OL brings significant improvements to the translation engines, in terms of translation quality and human effort reduction. Comparisons with other works in the literature also show that our adaptive, interactive systems are able to outperform the existing state-of-the-art.
5. We open-source all code developed in this work, in order to make research reproducible.

The rest of this manuscript is structured as follows: the related work is reviewed in Section 2. Next, Section 3 briefly introduces the NMT technology, while the interactive protocol for NMT is presented in Section 4. OL is described and applied together with INMT in Section 5. Section 6 describes the experimental setup of this work. Results are presented and discussed in Section 7. Finally, we conclude the work and trace future lines of work in Section 8.

2. Related work

This work puts together three thoroughly studied fields: neural machine translation, interactive machine translation and online learning. In this section, we briefly review the progress made in the last years in each one of these fields.

2.1. Neural machine translation

Although the first attempts of performing machine translation with neural networks date from long ago (Castaño and Casacuberta, 1997; Forcada and Ñeco, 1997), NMT only took off recently. Kalchbrenner and Blunsom (2013) reintroduced full neural MT, although the results were non-competitive with respect to classical PB-SMT systems. Nevertheless, in the next year, Cho et al. (2014) and Sutskever et al. (2014) proposed two similar sequence-to-sequence models, applied to the MT problem with encouraging results. These works were based on an encoder–decoder architecture, implemented with recurrent neural networks, with long short-term memory (LSTM) units (Hochreiter and Schmidhuber, 1997) or gated recurrent units (GRU) (Cho et al., 2014). From here, the NMT technology had a meteoric trajectory. Bahdanau et al. (2015) introduced the so-called attention model in the sequence-to-sequence framework. This allowed the system to selectively focus on parts of the input sequence, providing good results when modeling long sequences.

This attentional NMT system was the basis of many works, which aimed to tackle its main issues, namely the management of large vocabularies and the out-of-vocabulary problem (Jean et al., 2015; Luong et al., 2015). The most satisfactory solution, was the use of subword sequences instead of words (Luong and Manning, 2016; Sennrich et al., 2016). This has become a de facto standard in NMT.

NMT systems are typically trained by means of stochastic gradient descent (SGD), with a maximum likelihood objective (see Section 3). Nevertheless, some works explored alternative cost functions. Reinforcement learning (Shen et al., 2016) or minimum risk training (Wu et al., 2016) strategies have been applied to NMT systems, generally with positive results.

At this moment, the NMT technology has reached the translation industry, and some companies have already adopted it as translation engine (Crego et al., 2016; Wu et al., 2016).

Moreover, the encoder–decoder framework can be applied to many other problems apart from MT, generally with good results. Among these applications we can find image captioning (Xu et al., 2015), video captioning (Yao et al., 2015; Peris et al., 2016), parsing (Vinyals et al., 2015) or speech translation (Duong et al., 2016). Multi-task learning approaches take advantage from this versatility and are also obtaining encouraging results (Kaiser et al., 2017).

2.2. Interactive machine translation

Since Foster et al. (1997) introduced the IMT, this approach has been continuously revised and developed (Alabau et al., 2013; Barrachina et al., 2009; Casacuberta et al., 2009; Langlais et al., 2002; Macklovitch et al., 2005), demonstrating and improving its capabilities. Hence, the original IMT protocol has been extended and modified in several ways: (Alabau et al., 2011; Sanchis-Trilles et al., 2008) included multimodal

feedback, (Azadi and Khadivi, 2015; Cai et al., 2013; Green et al., 2014) improved the suffix generation, (González-Rubio et al., 2010) integrated of confidence measures in the interactive pipeline, etc.

Given the recent success of NMT, this technology has also be adapted to fit into the interactive framework (Knowles and Koehn, 2016; Peris et al., 2017b). Alternative technologies, such as translation memories, also were modified to allow interaction (Green et al., 2014). Other works aimed to build resource-agnostic IMT systems (Pérez-Ortiz et al., 2014).

A significant effort has also been spent in overcoming the tight left-to-right constraint of classical IMT systems. Marie and Max (2015) proposed a system based on touch-interactions, which allowed the user to select the correct parts of a hypothesis. Extending this work, Cheng et al. (2016) developed a pick-revise procedure for IMT, consisting in the selection by the user of the most critical part of a hypothesis and its correction. This pick-revise framework has been also applied to NMT systems (Hokamp and Liu, 2017). Related to this, González-Rubio et al. (2016), Domingo et al. (2018) and Peris et al. (2017b) allowed the selection of correct segments from translation hypotheses, which must be remain fixed along the IMT process.

It is worth noting a major difference between these last two works and the one developed by Cheng et al. (2016): while González-Rubio et al. (2016) and Peris et al. (2017b) demand perfect translations for a given sentence (as in a full post-editing setup), Cheng et al. (2016) accept some translation errors, sacrificing the final quality at the expense of a minor human effort (as in light post-editing).

2.3. Online learning in machine translation

Online learning is a paradigm thoroughly explored in the literature. In the field of MT, most works aimed to adapt a MT system to the user or to tailor it for a given document. The most clear example is the use of OL techniques for adjusting the weights of the log-linear model of PB-SMT. A significant number of algorithms have been applied to this task. The margin-infuse relaxed algorithm (MIRA) (Crammer and Singer, 2001) processes all samples one-by-one and it becomes especially useful when dealing with a large number features. It has been applied to PB-SMT with sparse features (Watanabe et al., 2007; Chiang, 2012; Green et al., 2013b).

Another common usage of OL in MT, is to perform user or domain adaptation of a system. Martínez-Gómez et al. (2012) studied several OL algorithms for adjusting the weights of a PB-SMT system during the post-editing phase. Similarly, Mathur et al. (2013) introduced an additional features, which allowed to take into account corrections done by the user. Closely related to this, Denkowski et al. (2014) implemented dynamic translation and language models which, together with the tuning of weights from the log-linear model, provided a reduction of the human effort required for post-editing the outputs of a system.

Beyond the tuning of the weights of the log-linear model, the re-estimation of the sub-models that conform PB-SMT systems via OL has also received attention. Many advances in this direction were achieved during the CasMaCat (Alabau et al., 2013) and MateCat (Federico et al., 2014) projects. Lagarda et al. (2015) adapted a general PB-SMT system to a specific domain, during the post-editing stage. This work was extended to the interactive framework by Ortiz-Martínez (2016).

Few works studied the application of OL techniques to NMT in the post-editing scenario. Almost simultaneously, Turchi et al. (2017) and Peris et al. (2017a) posed a similar scenario, in which an NMT system was refined with post-edited samples in order to perform domain adaptation. Both works used online SGD in the continuous learning phase. The NMT system was significantly improved in almost every case. Additionally, Peris et al. (2017a) considered alternative optimization methods, but they obtained poorer results than using traditional SGD. In this work, we extend the application of OL to INMT. Moreover, we study the effectiveness of OL for NMT in multiple and varied setups.

3. Neural machine translation

Statistical approaches to MT (Brown et al., 1993) aim to find the most likely sentence $\hat{y}_1^{\hat{I}} = \hat{y}_1, \ldots, \hat{y}_{\hat{I}}$ in the target language, given a sentence $x_1^J = x_1, \ldots, x_J$ in the source language:

$$\hat{y}_1^{\hat{I}} = \arg\max_{I, y_1^I} \Pr(y_1^I \mid x_1^J) \tag{1}$$

NMT systems directly model this posterior probability. Most NMT approaches follow an encoder–decoder paradigm: the encoder is a neural network which computes a compact representation of the input sentence. The decoder is another neural network that takes this representation and decodes it into a sentence in the target language. Such networks are typically recurrent neural networks with LSTM units or GRU. Nevertheless, other network architectures also fit in the encoder–decoder framework (Gehring et al., 2017; Vaswani et al., 2017). Our NMT system was inspired by Bahdanau et al. (2015), but we took into account recommendations given by Sennrich et al. (2017) and Britz et al. (2017).

Each source sentence $x_1, \ldots, x_J$ is processed as a sequence of words. It is inputed to the system and projected to a continuous space by means of a word embedding matrix. The sequence of word embeddings feeds a bidirectional (Schuster and Paliwal, 1997) LSTM network, which concatenates the hidden states from the forward and backward layers, producing a sequence of annotations $\mathbf{h}_1, \ldots, \mathbf{h}_J$.

The decoder is a conditional LSTM (cLSTM) network which takes into account the sequence of annotations together with the previously generated word (y_{t-1}). The cLSTM unit represents a novel extension of the conditional GRU (cGRU) with attention (Sennrich et al., 2017) to LSTMs.

As cGRUs, a cLSTM unit is composed of LSTM transition blocks together with an attention mechanism. Fig. 1 shows an illustration of our cLSTM cell.

In our case, we use two LSTM blocks. The first block, combines the word embedding of the previously generated word ($\mathbf{E}(y_{t-1})$) together with the hidden state from the second LSTM block at the previous time-step ($\mathbf{s}_{t-1}$), obtaining the intermediate representation $\mathbf{s}'_t$:

$$\mathbf{s}'_t = \mathrm{LSTM}_1(\mathbf{E}(y_{t-1}), \mathbf{s}_{t-1}) \tag{2}$$

The LSTM_1 function is defined according to the following equations[1] (Hochreiter and Schmidhuber, 1997; Gers et al., 2000):

$$\begin{aligned}
\mathbf{s}'_t &= \mathbf{o}'_t \odot \mathbf{c}'_t \\
\mathbf{c}'_t &= \mathbf{f}'_t \odot \mathbf{c}'_{t-1} + \mathbf{i}'_t \odot \tilde{\mathbf{c}}'_t \\
\tilde{\mathbf{c}}'_t &= \tanh(\mathbf{W}'_c \mathbf{E}(y_{t-1}) + \mathbf{U}'_c \mathbf{s}_{t-1}) \\
\mathbf{f}'_t &= \sigma(\mathbf{W}'_f \mathbf{E}(y_{t-1}) + \mathbf{U}'_f \mathbf{s}_{t-1}) \\
\mathbf{i}'_t &= \sigma(\mathbf{W}'_i \mathbf{E}(y_{t-1}) + \mathbf{U}'_i \mathbf{s}_{t-1}) \\
\mathbf{o}'_t &= \sigma(\mathbf{W}'_o \mathbf{E}(y_{t-1}) + \mathbf{U}'_o \mathbf{s}_{t-1})
\end{aligned}$$

where $\mathbf{E}$ is the target text word embedding matrix and $\mathbf{E}(y_{t-1})$ denotes the word embedding of the previously generated word (y_{t-1}). $\mathbf{i}'_t$, $\mathbf{o}'_t$ and $\mathbf{f}'_t$ are the input, output and forget gates, which control the information flow along the cell. $\mathbf{c}'_t$ and $\tilde{\mathbf{c}}'_t$ are the so-called cell and updated cell states, respectively. $\mathbf{W}'_c$, $\mathbf{U}'_c$, $\mathbf{W}'_f$, $\mathbf{U}'_f$, $\mathbf{W}'_i$, $\mathbf{U}'_i$, $\mathbf{W}'_o$ and $\mathbf{U}'_o$, are the trainable weight matrices. $\odot$ denotes element-wise multiplication and σ, logistic sigmoid activation function.

At each decoding time-step t, an attention mechanism weights every element from the sequence of annotations, according to the intermediate representations obtained in Eq. 2. A single-layer perceptron computes an alignment score, as in Bahdanau et al. (2015):

$$e_{jt} = \mathbf{w}^\top \tanh(\mathbf{W}_a \mathbf{s}'_t + \mathbf{U}_a \mathbf{h}_j) \tag{3}$$

[1] For notation simplicity, we omit the bias terms in all expressions.

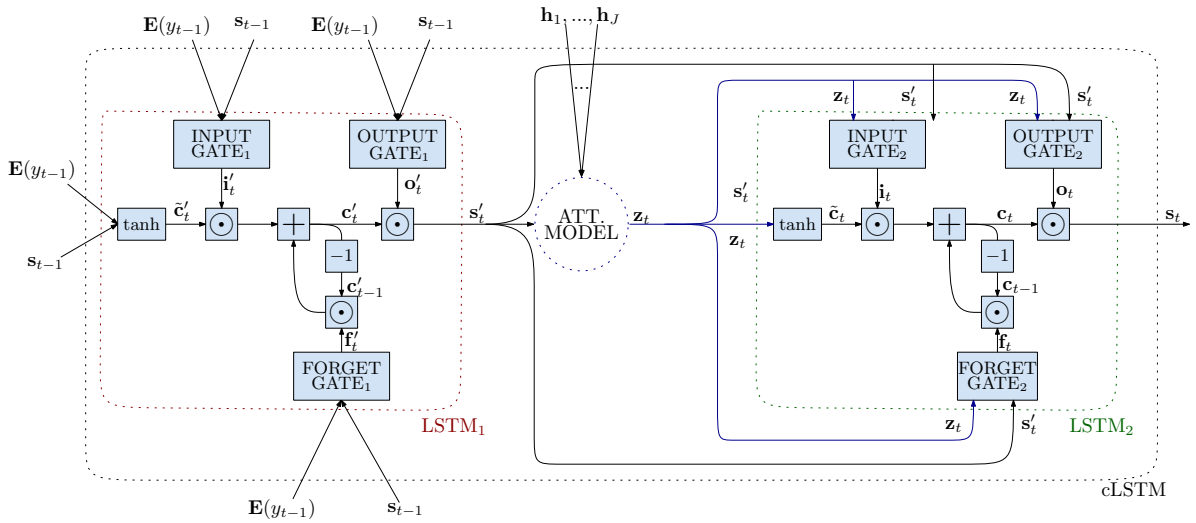


Figure 1: Conditional LSTM (cLSTM) cell, with attention, as used in the decoder. Dotted lines separate each one of its components: the first LSTM block (red), the attention mechanism (blue) and the second LSTM block (green). $\mathbf{E}(y_{t-1})$ is the word embedding of the previously generated word, $\mathbf{s}'_t$ and $\mathbf{s}_t$ are the hidden states of the LSTM blocks at the time-step t and $\mathbf{z}_t$ is the context vector computed by the attention model from the sequence of annotations $\mathbf{h}_1, \ldots, \mathbf{h}_J$ and $\mathbf{s}'_t$.

where $\mathbf{w}$, $\mathbf{W}_a$ and $\mathbf{U}_a$ are trainable parameters. This aligner is then followed by a softmax function, for obtaining normalized weights:

$$\alpha_{jt} = \frac{\exp\left(e_{jt}\right)}{\sum_k^J \exp\left(e_{kt}\right)}. \tag{4}$$

Finally, the context vector ($\mathbf{z}_t$) is computed as a weighted sum of the annotations:

$$\mathbf{z}_t = \sum_{j=1}^{J} \alpha_{jt} \mathbf{h}_j, \tag{5}$$

This context vector is the input to the second LSTM block, which also takes into account the intermediate representation $\mathbf{s}'_t$:

$$\mathbf{s}_t = \text{LSTM}_2(\mathbf{s}'_t, \mathbf{z}_t) \tag{6}$$

The LSTM_2 transition block is similar to the LSTM_1, but with different inputs:

$$\begin{aligned}
\mathbf{s}_t &= \mathbf{o}_t \odot \mathbf{c}_t \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \\
\tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_c \mathbf{z}_t + \mathbf{U}_c \mathbf{s}'_t) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{z}_t + \mathbf{U}_f \mathbf{s}'_t) \\
\mathbf{i}_t &= \sigma(\mathbf{W}_t \mathbf{z}_t + \mathbf{U}_t \mathbf{s}'_t) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o \mathbf{z}_t + \mathbf{U}_o \mathbf{s}'_t)
\end{aligned}$$

where, as above, $\mathbf{W}_c$, $\mathbf{U}_c$, $\mathbf{W}_f$, $\mathbf{U}_f$, $\mathbf{W}_t$, $\mathbf{U}_t$, $\mathbf{W}_o$ and $\mathbf{U}_o$ are the trainable weight matrices; $\mathbf{i}_t$, $\mathbf{o}_t$ and $\mathbf{f}_t$ are the input, output and forget gates; and $\mathbf{c}_t$ and $\tilde{\mathbf{c}}_t$ are cell and updated cell states.

The output of the decoder $\mathbf{s}_t$ is combined together with context vector $\mathbf{z}_t$ and the word embedding of the previously generated word $\mathbf{E}(y_{t-1})$ in a deep output layer (Pascanu et al., 2014), to obtain an L-sized intermediate representation $\mathbf{t}_t$:

$$\mathbf{t}_t = \tanh(\mathbf{s}_t \mathbf{W}_{t1} + \mathbf{z}_t \mathbf{W}_{t2} + \mathbf{E}(y_{t-1}) \mathbf{W}_{t3}) \tag{7}$$

where $\mathbf{W}_{t1}$, $\mathbf{W}_{t2}$ and $\mathbf{W}_{t3}$ are trainable weight matrices.

Finally, the probability of the word y_t at time-step t is defined as:

$$p(y_t \mid y_1^{t-1}, x) = \bar{\mathbf{y}}_t^\top \text{softmax}(\mathbf{V}\mathbf{t}_t) \tag{8}$$

where $\mathbf{V} \in \mathbb{R}^{|V_y| \times L}$ is a weight matrix, $|V_y|$ is the size of the target language vocabulary and $\bar{\mathbf{y}}_t \in [0,1]^{|V_y|}$ is the one-hot codification of word y_t.

3.1. Training and decoding

All model parameters θ (the weight matrices of the neural networks) are jointly estimated on a parallel corpus $\mathcal{S} = \{(x^{(s)}, y^{(s)})\}_{s=1}^{S}$, consisting of S sentence pairs. The training objective is to minimize a loss function ℓ_θ, typically the minus log-likelihood, over $\mathcal{S}$:

$$\begin{aligned} \widehat{\theta} &= \arg\min_\theta \ \ell_\theta(\mathcal{S}) = \\ &= \arg\min_\theta \sum_{s=1}^{S} \sum_{t=1}^{I_s} -\log(p_\theta(y_t^{(s)} \mid y_1^{t-1(s)}, x^{(s)})) \end{aligned} \tag{9}$$

where I_s is the length of the s-th target sentence and $y_1^{t-1(s)}$ denotes the s-th target sentence up to the position $t-1$.

A beam search method is used at decoding time, in order to find the most likely translation (Bahdanau et al., 2015; Sutskever et al., 2014).

4. Interactive neural machine translation

In the general framework of IMT (Barrachina et al., 2009), a source sentence x_1^J is inputted to the system, which produces a translation hypothesis $\hat{y}_1^{\hat{I}}$. Next, a human agent reviews the hypothesis and provides a feedback signal f. The IMT system produces a new translation hypothesis, considering this user feedback. It is expected that this new hypothesis is better than the previous one, as the system has more information. Then, a new iteration starts. This iterative procedure continues until the user accepts the hypothesis provided by the system. Fig. 2 shows an illustration of the IMT framework.

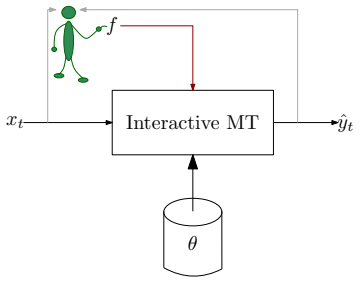


Figure 2: IMT framework. The MT system generates a translation hypothesis $\hat{y}_t$ of the source sentence x_t. The user reviews this hypothesis, and introduces a feedback signal f. In the next iteration, the system will consider f for generating a new (and hopefully better) hypothesis.

The feedback signal can range from a simple word correction to complex, indeterministic ones, such as an eye-gaze tracking, combined with a handwritten correction, as in Alabau et al. (2013). The features of the signal and its meaning determine the nature and behavior of the IMT system. In this work, we use keyboard and mouse to introduce feedback to the INMT system.

From a probabilistic point of view, the inclusion of the feedback signal affects Eq. (1), which now is also conditioned to f:

$$\hat{y}_1^{\hat{I}} = \arg\max_{I, y_1^I} \ \Pr(y_1^I \mid x_1^J, f) \tag{10}$$

In this work, we refine the prefix-based interactive approach presented by Peris et al. (2017b), who extended this IMT protocol to NMT. In this case, the feedback signal provided by the user contained the correction of the leftmost wrong word from the translation hypothesis, located at position t'. With this, the user inherently validated a translation prefix (up to t'). Hence, f can be seen as a validated prefix: $f = \hat{y}_1^{t'}$.

Next, the system provided an alternative hypothesis, which contained the validated prefix together with an alternative suffix. Therefore, Eq. (10) was reformulated as:

$$\hat{y}_1^{\hat{I}} = \arg\max_{I, y_{t'+1}^I} \Pr(y_{t'+1}^I \mid x_1^J, \hat{y}_1^{t'}) \tag{11}$$

which implies a search over the space of translations, but constrained by the validated prefix $\hat{y}_1^{t'}$.

The application of this expression to NMT implies a modification of the search method, for taking into account the validated prefix $\hat{y}_1^{t'}$. Therefore, the probability of a word (Eq. 8) is now computed as:

$$p(y_t \mid y_1^{t-1}, x_1^J, \hat{y}_1^{t'}) = \begin{cases} \delta(y_t, \hat{y}_{t'}), & \text{if } t \leq t' \\ \bar{\mathbf{y}}_t^\top \text{softmax}(\mathbf{V}\mathbf{t}_t) & \text{otherwise} \end{cases} \tag{12}$$

where $\delta(\cdot,\cdot)$ is the Kronecker delta. As in Eq. (8), $\bar{\mathbf{y}}_t$ is the one-hot codification of word y_t and $\text{softmax}(\mathbf{V}\mathbf{t}_t)$ is the output layer of the NMT system.

4.1. Vocabulary masking for character-level INMT

The INMT system developed by Peris et al. (2017b) performed all interactions at a word level. Nevertheless, it is interesting to enables the user to interact with the system at a character level. This makes possible a major granularity and a more natural interaction with the system. Most of the existing IMT tools already accept character-level interactions (e.g. CasMaCat).

In the field of NMT, to perform translations at character-level is a promising research direction (Costa-Jussà and Fonollosa, 2016; Chung et al., 2016). Unfortunately, the prohibitive decoding times of character-level NMT (Lee et al., 2016) prevent its direct usage in an interactive setup.

In this work, we propose a simple, yet effective way for interacting with the INMT system at character level. The feedback signal provided by the user is at character-level. For the sake of simplicity, we stick to prefix-based interaction. That is, the user will correct the hypotheses from the left to the right. Therefore, the user inputs the leftmost wrong character of a hypothesis. Nevertheless, the same method is extensible to other interactive protocols (e.g. segment-based interaction (Peris et al., 2017b)).

As before, the system must produce a hypothesis compatible with the user feedback. In this case, the user introduced a character correction in the u-th position of the t'-th word. Therefore, the validated prefix are all words up to position $t'-1$ together with the validated part of the t'-th word:

$$f = \hat{y}_0^{t'-1}, \hat{y}_{t'_1^u}$$

where $\hat{y}_0^{t'-1}$ is the sequence of validated words, up to word in position $t'-1$, $\hat{y}_{t'_1^u}$ is the correct part of the word $\hat{y}_{t'}$ together with the corrected character position u.

When processing this signal, we may need to generate a word constrained by the prefix $\hat{y}_{t'_1^u}$ or not. The latter case is handled by the classical NMT system without modifications.

For tackling the first case, we create a mask $\mathbf{m}_u$ of the target vocabulary according to the user prefix $\hat{y}_{t'_1^u}$. Therefore, $\mathbf{m}_u \in [0,1]^{|V_y|}$ is a vocabulary-sized binary vector, in which each position is set to 1 if the corresponding word in the vocabulary is compatible with the user prefix and to 0 otherwise. If there are no compatible words with the validated prefix, we apply forced decoding to this prefix and continue the process with the unconstrained vocabulary. Fig. 3 shows an example of this masking strategy.

The word probability expression (Eq. (12)) is then computed as:

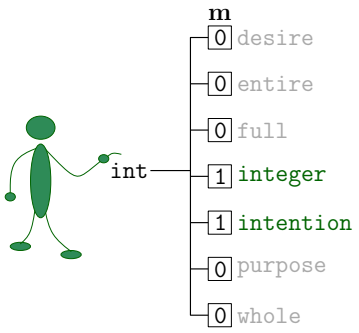


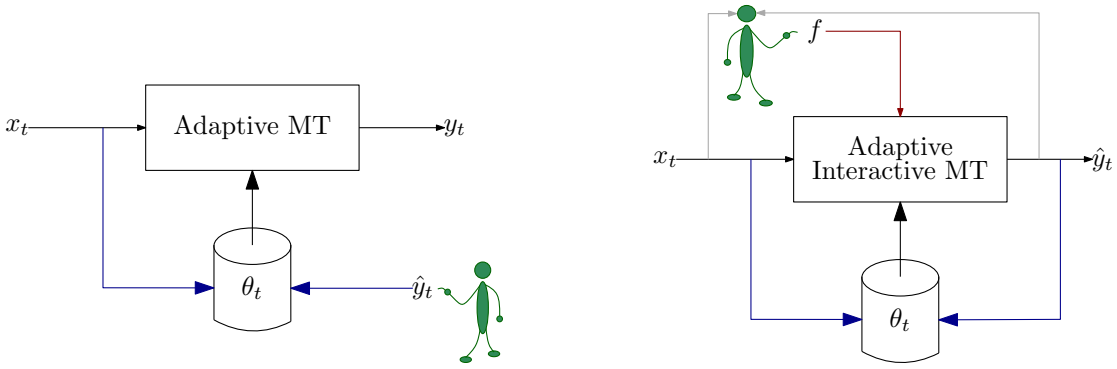
Figure 3: Constraining the vocabulary for character-level interaction. For this example, we assume a vocabulary of 7 words. The user introduces a valid prefix consisting one or more characters (`int`). For predicting the next word, the system computes a compatibility mask $\mathbf{m}$, which filters those words that are incompatible with the given prefix (in gray). In this case, the compatible words (in green) are `integer` and `intention`.

$$p(y_t \mid y_0^{t-1}, x_1^J, \hat{y}_1^{t'-1}, \hat{y}_{t'\,1}^{\,u}) = \begin{cases} \delta(y_t, \hat{y}_{t'}), & \text{if } t < t' \\ \mathbf{m}_u^\top \bar{\mathbf{y}}_t^\top \text{softmax}(\mathbf{V}\mathbf{t}_t), & \text{if } t = t' \\ \bar{\mathbf{y}}_t^\top \text{softmax}(\mathbf{V}\mathbf{t}_t) & \text{otherwise} \end{cases} \tag{13}$$

With this strategy, we get the benefits of character-level interaction while maintaining the decoding speed of (sub)word-level NMT. Moreover, since we keep the probabilities of each compatible word, is straightforward to implement additional features to the system, such as word completion.

It is also remarkable that this vocabulary masking strategy can help the system to disambiguate words. For instance, by looking at Fig. 3, if we filter the vocabulary, the system must choose between `integer` and `intention`. This reduces the possible ambiguity with other vocabulary words, such as `entire`, `full` or `whole`.

5. Online learning in NMT post-editing



(a) Machine translation post-editing.

(b) Interactive machine translation.

Figure 4: Online learning in MT post-editing and IMT. The system translates a source sentence x_t, producing a hypothesis y_t. This hypothesis is corrected. The corrected hypothesis $\hat{y}_t$ is used, together with x_t by the system to modify its models (parametrized by θ_t). The difference between classical post-editing and IMT is the way that the hypothesis is corrected.

Under an online learning paradigm, data comes available sequentially and models are updated incrementally. The typical post-editing or IMT scenario perfectly matches with these stages:

1. A new source sentence x_t comes to the system.
2. The system produces a translation hypothesis y_t.
3. A human agent revises this hypothesis and corrects the errors made by the system or interactively translates the source sentence. This generates a correct translation $\hat{y}_t$.
4. The system uses the corrected sample to adapt its models.

Following this procedure, we can build adaptive MT systems, able to take into account corrections made by the humans. Fig. 4 illustrates the adaptive MT framework, either with IMT or classical post-editing.

Adaptive NMT systems have interesting applications in the translation industry. The NMT technology has good results if the training corpora are large enough (Koehn and Knowles, 2017). But to acquire high amounts of parallel corpora is an expensive process. Moreover, for building a NMT system for a given domain, we need data from this domain, which can be difficult to obtain. A common approach is to train a system on a large, general corpus and fine-tune it with in-domain data. Nevertheless, this is may also be infeasible; for instance, if the domain is unknown.

Therefore, continuous learning from MT post-edits or IMT constitute techniques that can be exploited for adapting a MT system to different domains or styles. They are orthogonal to other adaptation techniques,

like fine-tuning with in-domain data. In this work, we apply OL to NMT systems for performing domain adaptation of general NMT systems. Moreover, we also study the power of OL for refining an NMT system already trained with in-domain data.

The most common training procedure of neural networks is SGD (Section 3.1), which can directly be applied in an online way. Therefore, the application of OL to NMT becomes natural. Online adaptation of NMT systems can be performed with the same optimizers than used during (mini-batch) training, but sample-to-sample.

For a training sample $(x_t, \hat{y}_t)$, SGD updates the parameters following the direction of the gradient of the objective function ℓ (Eq. (9)) with respect to the weights θ_t:

$$\Delta\theta_t = -\rho \, \nabla \ell_{\theta_t}(x_t, \hat{y}_t) \tag{14}$$

where $\nabla \ell_{\theta_t}$ is the gradient of ℓ with respect to θ_t and ρ is a learning rate that controls the step size.

This update rule relies on a careful choice of ρ. A significant effort has been spent in the literature trying to minimize the critical importance of the learning rate choice. Therefore, the so-called adaptive SGD algorithms try to overcome this dependence by dynamically computing the learning rate.

Among the most common adaptive optimizers, we find AdaGrad (Duchi et al., 2011), which updates the weights according to the sum of the squares of the past gradients; Adam (Kingma and Ba, 2014) which computes decaying averages for the past gradients and past squared gradients or Adadelta (Zeiler, 2012), which updates the parameters according to the root mean square (RMS) of the gradients and corrects these updates according to the RMS of the previous update.

Nevertheless, in our scenario of online learning in NMT, the usage of adaptive SGD algorithms does not completely alleviate the learning rate tuning. We found (Section 7.1) that a correct choice of the learning rate is extremely important, even for adaptive SGD optimizers, to make them properly work.

6. Experimental framework

We conducted an exhaustive experimentation in order to assess the effectiveness of our proposals. This section details the experimental setup, evaluation metrics, simulation procedure, corpora and the translation systems.

6.1. Corpora

We evaluated our models in 5 tasks of different complexity and nature. For all corpora, we performed the translation from English to German and from English to French. First, we used two corpora extensively used in the IMT literature: The XRCE and EU corpora (Barrachina et al., 2009). The former consists of printer manuals from Xerox printers and the latter is a collection of proceedings collected from the Bulletin of the European Union. For these tasks, we used the default data splits. We also tested our proposals with data from WMT'17 (Bojar et al., 2017a), using the Europarl and the UFAL corpora. Europarl (Koehn, 2005) collects the proceedings from the European Parliament. For the sake of comparison with Ortiz-Martínez (2016), we used the `newstest2012` and the `newstest2013` as development and test partitions respectively. The UFAL medical corpus[2] contains data crawled from several medical collections, collected during the European project *Health in my Language* (Bojar et al., 2017c). We used the development and test data from the Khresmoi project (Libovický et al., 2016). Finally, the TED task (Mauro et al., 2012) refers to the translation of TED talks and it has also been used in IMT and online learning works. We used the standard `dev2010` and `tst2010` partitions for development and test, respectively.

All corpora were tokenized using the Moses scripts. For training the NMT systems, we applied joint byte-pair-encoding (BPE) (Sennrich et al., 2016) to all corpora, using 32,000 merge operations. Table 1 shows the main figures of each corpus, after tokenization. For the test set, we show additional metrics, aiming to obtain an estimation of the potential efficacy of the OL process:

[2] https://ufal.mff.cuni.cz/ufal_medical_corpus

Repetition rate (RR) (Bertoldi et al., 2013): measures the repetitiveness of a document. It is computed as the rate of non-singleton n-grams (with n from 1 to 4). The rates are obtained through a sliding window of $1,000$ words and geometrically averaged.

Restricted repetition rate (RRR) (Ortiz-Martínez, 2016): RR is unable to capture whether a specific n-gram from the text to translate was in the document used for training the models. Therefore, it may be insufficient to properly estimate the potential of OL. The RRR aims to overcome this issue by computing the RR on those n-grams from the test not contained in the training data.

Unseen n-gram fraction (UNF) (Ortiz-Martínez, 2016): ratio of unseen n-grams in the test document. As in the RR, we consider n-grams from order 1 up to 4.

The effectiveness of OL can be estimated beforehand by paying attention to these three metrcs metrics (Ortiz-Martínez, 2016). OL will likely be more effective for documents tasks with high values of RR, RRR and UNF.

Table 1: Main figures of the XRCE, EU, UFAL, Europarl and TED corpora. $|S|$, $|T|$ and $|V|$, RR [%] account for number of sentences, number of tokens, vocabulary size, repetition rate, restricted repetition rate unseen n-gram fraction, respectively. RR, RRR and UNF were computed after the BPE process. k and M stand for thousands and millions.

		Training			Development			Test					
		$\|S\|$	$\|T\|$	$\|V\|$	$\|S\|$	$\|T\|$	$\|V\|$	$\|S\|$	$\|T\|$	$\|V\|$	RR [%]	RRR [%]	UNF [%]
XRCE	De	$50k$	$531k$	$23k$	964	$10k$	$2k$	995	$12k$	$2k$	23.6	17.4	25.7
	En		$587k$	$11k$		$11k$	$1k$		$12k$	$2k$	28.0	16.0	9.1
	Fr	$52k$	$676k$	$16k$	994	$12k$	$2k$	984	$12k$	$2k$	27.9	18.3	31.5
	En		$615k$	$15k$		$11k$	$2k$		$11k$	$2k$	26.9	18.3	12.6
EU	De	$222k$	$5.4M$	$109k$	400	$10k$	$3k$	800	$19k$	$5k$	9.3	0.0	3.0
	En		$5.7M$	$42k$		$10k$	$2k$		$20k$	$4k$	12.3	0.7	3.5
	Fr	$215k$	$6.2M$	$58k$	400	$12k$	$3k$	800	$24k$	$4k$	14.0	0.0	0.0
	En		$5.2M$	$50k$		$10k$	$3k$		$20k$	$4k$	12.3	0.0	2.4
UFAL	De	$3.0M$	$109M$	$1.6M$	500	$10k$	$3k$	$1,000$	$19k$	$3k$	10.7	3.6	4.2
	En		$116M$	$671k$		$10k$	$3k$		$21k$	$4k$	13.4	0.0	0.0
	Fr	$2.8M$	$125M$	$647k$	500	$12k$	$3k$	$1,000$	$26k$	$5k$	16.6	5.9	3.3
	En		$109M$	$655k$		$10k$	$3k$		$21k$	$4k$	13.1	0.0	4.3
Europarl	De	$1.9M$	$50M$	$393k$	$3,003$	$73k$	$14k$	$3,000$	$65k$	$12k$	13.4	8.9	18.4
	En		$53M$	$123k$		$73k$	$10k$		$63k$	$10k$	15.0	9.4	15.4
	Fr	$2.0M$	$61M$	$153k$	$3,003$	$82k$	$11k$	$3,000$	$74k$	$11k$	12.6	8.1	7.5
	En		$56M$	$134k$		$71k$	$10k$		$65k$	$10k$	14.2	6.6	7.9
TED	De	$133k$	$2.4M$	$102k$	883	$19k$	$4k$	$1,565$	$30k$	$5k$	11.2	4.7	3.7
	En		$2.6M$	$50k$		$20k$	$3k$		$32k$	$4k$	14.6	7.0	4.8
	Fr	$107k$	$2.2M$	$58k$	934	$20k$	$4k$	$1,664$	$34k$	$5k$	14.8	4.3	7.2
	En		$2.1M$	$47k$		$20k$	$3k$		$32k$	$4k$	14.8	5.2	6.3

6.2. Evaluation

The ultimate goal of this work is to reduce the human effort required in the MT supervision process. Therefore, we must assess either the translation quality of the MT system and, more importantly, this human effort. We evaluated both, translation quality and human effort.

6.2.1. Translation quality metrics

Quality assessment of MT is an open problem. The main goal of an automatic metric is to achieve a high correlation with human judgment of the quality of a translation (Bojar et al., 2017b). For doing this, the MT translation hypotheses are usually compared to a reference translation. We evaluated the quality of the MT systems according to two common metrics:

Translation Edit Rate (TER) (Snover et al., 2006): minimum number of edit operations required for converting the translation hypothesis into the reference, divided by the number of reference words. The edit operations considered are insertion, deletion, replacement and word sequences shifting. The minimum number of operations is obtain by means of dynamic programming. Following Zaidan and Callison-Burch (2010), we use the TER as a representation of human-targeted TER, considering the reference sentences as human post-edited versions of the MT hypotheses. This gives us a broad approximation of the effort required for post-editing a translation hypothesis.

BiLingual Evaluation Understudy (BLEU) (Papineni et al., 2002): geometric mean of n-gram matchings between hypothesis and reference, modified by a brevity penalty. We use BLEU for having additional insights about the complexity of the task and the performance of the systems in terms of translation quality.

6.2.2. Human effort metrics

Human agents are involved during the IMT process, therefore, we must estimate the effort spent. We are aware that the correct way of doing this is to conduct an experimentation with real users. Nevertheless, prior to this, we need to automatically assess the effort required in the IMT scenario, in order to construct and develop efficient IMT systems. Therefore, we assumed that the reference sentences are the translations desired by the user. We estimated the human effort required according to:

Keystroke and mouse-action ratio (KSMR) (Barrachina et al., 2009): number of keystrokes plus number of mouse actions required in the IMT process, divided by the number of characters of the reference. Therefore, the lower KSMR, the better. If the user is correcting contiguous characters, no mouse action is needed. An additional mouse action that accounts for the acceptation of a hypothesis is added to each sentence.

6.3. Simulation protocol

Due to the aforementioned reasons, we evaluated our proposals with simulated users, assuming that the reference sentences are the translations that the users have in mind for each source sentence.

In the case of IMT, since we use the prefix-based protocol, we searched for the leftmost wrong character of a translation hypothesis, comparing it with the reference. Next, we introduced the correct character, inputting this feedback signal to the system. With this, the IMT engine produced a new hypothesis that takes into account the user feedback (Eq. (13)). This process continued until translation hypothesis and reference matched. The user would then validate the hypothesis and finishing the process.

In the case of the post-editing scenario, we translated a source sentence. Next, the user would edit the translation hypothesis, producing the desired translation. We directly used the reference sentences as those desired translations.

Finally, we apply OL (as described in Section 5) using these edited sentences, performing the adaptation sample to sample. That is, before starting the translation of the next sentence, the models are updated according the previous sample in both the post-editing and IMT scenarios.

6.4. Machine translation systems

We built our NMT systems using NMT-Keras[3] (Peris and Casacuberta, 2018b). Our systems can be implemented using either Tensorflow (Abadi et al., 2016) or Theano (Theano Development Team, 2016). In our experiments, used the latter framework. Taking advantage of the extensive experimentation conducted by Britz et al. (2017), we set the dimensions of encoder, decoder, attention model and word embedding to 512. Due to hardware restrictions, we used a single layer for encoder and decoder. The encoder was a LSTM network, while the decoder was made of cLSTM units (see Section 3).

[3]Implementations of all models, algorithms and search methods are publicly available at https://github.com/lvapeab/nmt-keras/tree/interactive_NMT.

The learning algorithm for all base NMT systems was Adam (Kingma and Ba, 2014), with a learning rate of 0.0002, as in Wu et al. (2016). We clipped the L_2 norm of the gradients to 5, in order to avoid the exploding gradient effect (Pascanu et al., 2012). The batch size was set to 50 and the beam size to 6. Since we are in an interactive scenario, we discarded the usage of model ensembles at this point of the study, fostering decoding and retraining speed. We trained all models on a single GeForce GTX 1080 GPU.

As regularization strategies, we applied Gaussian noise to the weights during training (Graves, 2011) and batch normalizing transform (Ioffe and Szegedy, 2015). We early stopped the training according to the BLEU of the development set. BLEU was checked each $5,000$ updates and the patience was set to 20.

For the sake of comparison, we also include results of PB-SMT. We estimated PB models with the standard setup of Moses (Koehn et al., 2007): 5-gram language model with modified Kneser-Ney smoothing (Kneser and Ney, 1995), built with SRILM (Stolcke, 2002). The phrase table was obtained using symmetrised word alignments, computed by GIZA++ (Och and Ney, 2002). The model also include distortion and reordering models. The weights of the log-linear model was optimized using Minimum Error Rate Training (MERT) (Och, 2003).

7. Results and discussion

In this section, we show and discuss the results obtained. For all result in this paper, we computed confidence intervals (at a 95% level). Confidence intervals were obtained by means of pairwise bootstrap resampling (Koehn, 2004).

We posed three different experimental conditions, varying the amount and type of training data available. In each case, we compared the performance of adaptive versus static systems, in translation post-editing and IMT. The evaluation was always carried out on the test set of each task. Continuous learning techniques were also applied exclusively on the same test data. Therefore, the three cases we studied are:

1. Exclusive availability of in-domain data.
2. Lack of in-domain data.
3. Availability of in-domain and out-of-domain data.

7.1. Scenario #1: Availability of in-domain data

We first assume that we have enough in-domain data for training a system. Therefore, we followed the traditional pipeline in MT: we trained translation systems using corpora from a given domain and translated documents from the same domain. In this case, online learning techniques aim to refine each system to the test documents.

For the sake of comparison, we first evaluated PB-SMT and NMT systems for performing classical translation post-editing. We mostly focused on TER, using it as an estimation of the human effort required for post-editing the output of a MT system (see Section 6.2). The results of PB-SMT and NMT systems are shown in Table 2.

In general, the NMT approach performed slightly better than PB-SMT systems, achieving significant TER improvements for five language pairs. Only in two cases PB-SMT systems significantly outperformed NMT. In the rest of cases, differences were non-significant. Hence, human effort required for post-editing the NMT outputs was usually lower than the required for post-editing PB-SMT systems, although such differences were small. BLEU behaviored similarly to TER: in 6 cases, NMT obtained significantly better translation hypotheses than PB-SMT. Only in one language pair, the PB-SMT system was able to significantly outperform NMT.

Next, we move to interactive machine translation. Table 3 shows the performance in % KSMR of the INMT systems. We also compare these results with the best results obtained in the literature for each task. All IMT systems from the literature were PB-SMT.

In terms of effort, INMT systems were substantially better than all state-of-the-art systems. In all cases, the effort was greatly reduced: from 7.9 to 16.9 points in KSMR percentage. This approximately accounts from a 20% to a 36% relative improvement.

Table 2: Results of translation quality for all tasks in terms of TER [%] and BLEU [%]. We compare PB-SMT and NMT systems. Results that are statistically significantly better for each task and metric are boldfaced.

		TER [%]		BLEU [%]	
		PB-SMT	NMT	PB-SMT	NMT
XRCE	En→De	62.5 ± 1.0	64.1 ± 1.1	24.7 ± 0.9	24.4 ± 1.1
	En→Fr	$\mathbf{49.9 \pm 1.0}$	54.0 ± 1.1	$\mathbf{37.1 \pm 0.9}$	34.7 ± 1.2
EU	En→De	54.1 ± 1.0	54.6 ± 1.0	35.3 ± 1.1	35.8 ± 1.1
	En→Fr	41.5 ± 0.8	$\mathbf{39.3 \pm 0.8}$	47.1 ± 0.9	$\mathbf{50.0 \pm 0.9}$
UFAL	En→De	62.4 ± 0.5	$\mathbf{56.5 \pm 0.6}$	17.3 ± 0.5	$\mathbf{24.2 \pm 0.5}$
	En→Fr	47.5 ± 0.5	$\mathbf{46.4 \pm 0.6}$	35.0 ± 0.5	$\mathbf{37.2 \pm 0.6}$
Europarl	En→De	$\mathbf{62.2 \pm 0.3}$	63.1 ± 0.4	19.5 ± 0.3	20.0 ± 0.3
	En→Fr	56.1 ± 0.3	$\mathbf{55.0 \pm 0.3}$	26.5 ± 0.3	$\mathbf{27.8 \pm 0.3}$
TED	En→De	58.4 ± 0.5	$\mathbf{55.5 \pm 0.6}$	20.3 ± 0.4	$\mathbf{24.5 \pm 0.5}$
	En→Fr	51.4 ± 0.5	51.5 ± 0.5	29.9 ± 0.5	$\mathbf{32.1 \pm 0.5}$

Table 3: Effort required by interactive NMT systems (INMT) compared to the state of the art, in terms of KSMR [%]. ∇ represents absolute decrements in KSMR percentage. Results that are statistically significantly better for each task and metric are boldfaced. † refers to Ortiz-Martínez (2016), ‡ to Barrachina et al. (2009) and $-$ indicates missing results in the literature.

		KSMR [%]		
		INMT	Best in literature	∇
XRCE	En→De	$\mathbf{32.2 \pm 0.7}$	$40.1 \pm 1.2^\dagger$	7.9
	En→Fr	$\mathbf{27.5 \pm 0.6}$	$35.8 \pm 1.3^\dagger$	8.3
EU	En→De	$\mathbf{19.8 \pm 0.5}$	$30.5 \pm 1.1^\ddagger$	10.7
	En→Fr	$\mathbf{16.3 \pm 0.4}$	$25.5 \pm 1.1^\ddagger$	9.2
UFAL	En→De	22.5 ± 0.3	$-$	$-$
	En→Fr	19.7 ± 0.3	$-$	$-$
Europarl	En→De	$\mathbf{32.3 \pm 0.2}$	$49.2 \pm 0.4^\dagger$	16.9
	En→Fr	$\mathbf{29.8 \pm 0.2}$	$44.4 \pm 0.5^\dagger$	14.6
TED	En→De	28.0 ± 0.3	$-$	$-$
	En→Fr	26.8 ± 0.3	$-$	$-$

This improvement of INMT systems over classical PB-SMT systems was also reported by Knowles and Koehn (2016) and Peris et al. (2017b), who found that the INMT technology reacted much better to user feedback than interactive PB-SMT. Due to these large differences, in the rest of this work, we focus only on neural systems.

7.1.1. Translation post-editing with online learning

Now, we study the application of online learning during the post-editing stage. We built adaptive NMT systems, able to continuously learn from previous errors. Hopefully, this will lead to better systems, with higher translation quality and with a subsequent decrease of the effort required for correcting their outputs.

As we have available validation data, we conducted an exploration of the best SGD optimizer and its hyperparamters for each task. We performed a grid search over the validation set and chose the configurations which obtained the lowest TER values. We compared vanilla SGD, Adagrad, Adadelta and Adam algorithms. We left the learning rate as the only tunable hyperparameter. The rest were fixed to default. We explored learning rates in the values: $b \cdot 10^e, b \in \{1, 5\}, e \in \{1, -1, -2, -3, -4, -5, -6\}$.

Table 4 shows the best configuration obtained for each task. Adadelta outperformed other algorithms, for most tasks. This finding is contrary to the observed by Turchi et al. (2017), who found that the most effective algorithm was vanilla SGD. We conjecture that these differences are due to the fact that they fixed the learning rate of all dynamic optimizers to 0.01, which may be unsuitable in some cases.

In our experimentation, Adadelta was the most stable optimizer, obtaining the best results using learning rates of 0.1 or 0.5. Vanilla SGD also performed well, but we found it to be more unstable than Adadelta: it was harder to find an adequate learning rate and it performed worse than Adadelta in many cases. On the other hand, we found that Adagrad and Adam were excessively aggressive to be useful for adaptive NMT systems: they required very low learning rates, otherwise, they completely distort the model. And even with low learning rates, they always performed worse than Adadelta or plain SGD.

Table 4: Best online SGD optimizer for each task.

		Algorithm	ρ
XRCE	En→De	Adadelta	0.1
	En→Fr	Adadelta	0.1
EU	En→De	SGD	10^{-5}
	En→Fr	Adadelta	0.1
UFAL	En→De	Adadelta	0.1
	En→Fr	Adadelta	0.1
Europarl	En→De	SGD	0.005
	En→Fr	Adadelta	0.1
TED	En→De	Adadelta	0.1
	En→Fr	SGD	0.01

Table 5 shows the effect of including OL in the post-editing process. In almost every case, the effort required, measured in terms of TER, is improved, yielding significant reductions in all tasks but EU. BLEU is consequently improved, following the TER trend.

Table 5: Translation results in terms of TER [%] and BLEU [%], of an adaptive NMT system (OL-NMT), compared to static NMT. ∇ and Δ represent absolute decrements and increments in terms of percentage for the corresponding metric of the online NMT system with respect the static one. Results that are statistically significantly better for each task and metric are boldfaced.

		TER [%]			BLEU [%]		
		NMT	OL-NMT	∇	NMT	OL-NMT	Δ
XRCE	En→De	64.1 ± 1.1	$\mathbf{60.0 \pm 1.1}$	4.1	24.4 ± 1.1	$\mathbf{28.9 \pm 1.2}$	4.5
	En→Fr	54.0 ± 1.0	$\mathbf{48.7 \pm 1.1}$	5.3	34.7 ± 1.2	$\mathbf{40.3 \pm 1.3}$	5.6
EU	En→De	54.6 ± 1.0	53.8 ± 1.0	0.8	35.8 ± 1.1	36.1 ± 1.1	0.3
	En→Fr	39.3 ± 0.8	39.3 ± 0.8	0.0	50.0 ± 0.9	50.2 ± 0.9	0.2
UFAL	En→De	56.5 ± 0.6	$\mathbf{53.7 \pm 0.6}$	2.8	24.2 ± 0.5	$\mathbf{26.0 \pm 0.6}$	1.8
	En→Fr	46.4 ± 0.6	$\mathbf{41.7 \pm 0.6}$	4.7	37.2 ± 0.6	$\mathbf{41.9 \pm 0.6}$	4.7
Europarl	En→De	63.1 ± 0.4	$\mathbf{60.4 \pm 0.4}$	2.7	20.0 ± 0.3	$\mathbf{22.8 \pm 0.3}$	2.8
	En→Fr	55.0 ± 0.3	$\mathbf{53.4 \pm 0.3}$	1.6	27.8 ± 0.3	$\mathbf{29.7 \pm 0.3}$	1.9
TED	En→De	55.5 ± 0.6	$\mathbf{50.8 \pm 0.5}$	4.7	24.5 ± 0.5	$\mathbf{25.5 \pm 0.5}$	1.0
	En→Fr	51.5 ± 0.5	$\mathbf{50.5 \pm 0.5}$	1.0	32.1 ± 0.5	32.9 ± 0.5	0.7

The largest improvements were obtained in the XRCE task, with gains of 4.1 and 5.3 TER points. This was due to the high RR, RRR and UNF values of this task (Table 1), being especially adequate for incremental learning. We also observed significant gains in the UFAL, TED and Europarl tasks.

On the other hand, in the EU task, the adaptive system achieved very little differences with respect to the static one. The low values of RR, RRR (almost zero) and UNF explain this little enhancement. Given the lack repetitiveness of the document, an online system is unable to exploit the recently learned knowledge.

Compared to the existing literature, Ortiz-Martínez (2016), applied OL techniques to a PB-SMT system, for the XRCE and Europarl tasks. In the first case, they obtained large improvements in terms of BLEU (11.5 and 8.4, for En→De and En→Fr, respectively). Nevertheless, their baseline systems performed worse than ours. Including OL, they achieved a similar performance for the XRCE task than our online NMT systems. In the Europarl case, they improved their static system by 1.0 and 1.4 BLEU points, for En→De and En→Fr.

But in this case, their static systems were much worse than ours (13.1/21.2, for En→De/En→Fr). NMT systems perform better than PB-SMT if they have a large amount of training data (Koehn and Knowles, 2017), as in this case. Even though, OL was able to refine a strong NMT system trained with a large corpus.

7.1.2. Interactive machine translation with online learning

Table 6: KSMR [%] effort required for adaptive NMT systems (OL-INMT) compared to static INMT and the state-of-the-art IMT adaptive systems (based on PB-SMT). Results that are statistically significantly better for each task and metric are boldfaced. † refers to Ortiz-Martínez (2016) and – indicates missing results in the literature.

		KSMR [%]		
		INMT	OL-INMT	Best in literature
XRCE	En→De	32.2 ± 0.7	$\mathbf{27.9 \pm 0.7}$	$37.0 \pm 1.3^{\dagger}$
	En→Fr	27.5 ± 0.6	$\mathbf{22.5 \pm 0.6}$	$30.3 \pm 1.2^{\dagger}$
EU	En→De	19.8 ± 0.5	19.5 ± 0.5	–
	En→Fr	16.3 ± 0.4	16.2 ± 0.4	–
UFAL	En→De	22.5 ± 0.3	$\mathbf{21.5 \pm 0.3}$	–
	En→Fr	19.7 ± 0.3	$\mathbf{17.7 \pm 0.3}$	–
Europarl	En→De	32.9 ± 0.2	$\mathbf{29.0 \pm 0.2}$	$48.0 \pm 0.5^{\dagger}$
	En→Fr	29.8 ± 0.2	$\mathbf{28.0 \pm 0.2}$	$43.2 \pm 0.5^{\dagger}$
TED	En→De	28.0 ± 0.3	27.5 ± 0.3	–
	En→Fr	26.8 ± 0.3	$\mathbf{26.2 \pm 0.3}$	–

Next, we move towards the deployment of adaptive, interactive NMT systems. We used the same configuration as for post-editing. Table 6 shows the effect of adding OL to INMT systems. Moreover, we show other results obtained in the literature.

We found that adaptive INMT systems outperformed static ones. As in post-editing, most of these differences were significant. Again, the XRCE task is the most benefited by OL, but we also obtained especially good results in the Europarl corpora. Besides their RRR and UNF values, it should also be noticed that the Europarl test documents had more samples than others. Therefore, the INMT system was benefited from a longer process of adaptation.

The EU task had the same behavior than in post-editing (Table 5): because of the lack of repetitiveness, we obtained marginal and non-significant improvements.

Compared to the literature (Ortiz-Martínez, 2016), we obtained similar gains in terms of KSMR for the XRCE task (around 5 KSMR points). In the case of Europarl, we obtained higher KSMR decreases: 3.9/1.8 against 1.2/1.2, for the En→De and En→Fr language pairs, respectively. Moreover, the large advantage in KSMR that INMT systems had with respect PB-SMT models is maintained in the online version.

7.2. Scenario #2: Lack of in-domain data

In this second scenario, we assume that we have no in-domain training data available. This can be the case of a system trained with data from a general domain, but having to translate documents from a different (and potentially unknown) domain. We take advantage of OL for performing domain adaptation on-the-fly, from the general to the test domain. We expect to obtain better system hypotheses as the post-editing, and inherently the online learning processes go on. The refinement of the system will hopefully entail a decrease of the human effort required for post-editing the upcoming samples.

We took as general system the one trained with the Europarl corpus. In order to work with the same vocabulary, we applied the same BPE segmentation to all in-domain sets. Table 7 shows the vocabulary coverage of the NMT system and each one of the in-domain documents and the RRR and UNF metrics for each task. All values were computed according to the BPE version of each test document.

Table 7: Vocabulary coverage (C),restricted repetition rate (RRR) and unseen n-gram fraction (UNF) with respect to the out-of-domain corpus (Europarl). We applied the BPE codification learned for the Europarl corpus to each document.

		Training	Development	Test		
		C	C	C	RRR [%]	UNF [%]
XRCE	De	98.7	99.8	99.5	26.8	12.9
	En	99.7	99.9	99.9	27.7	11.7
	Fr	98.2	97.5	97.4	28.6	8.8
	En	98.5	97.2	97.2	33.1	8.4
EU	De	98.2	99.9	99.9	7.6	19.5
	En	95.1	99.7	99.6	8.1	17.5
	Fr	97.5	98.0	98.6	9.9	15.3
	En	96.0	97.6	98.4	8.2	17.2
UFAL	De	92.2	99.8	99.7	7.4	8.4
	En	85.9	99.8	99.8	13.4	7.8
	Fr	91.3	99.7	99.7	12.6	7.1
	En	92.5	99.7	99.8	10.5	7.6
TED	De	98.1	99.8	99.9	4.7	5.4
	En	99.2	99.9	99.9	11.0	4.5
	Fr	97.9	98.7	99.0	6.4	4.0
	En	98.3	98.7	98.8	8.9	3.6

The vocabulary coverage was extremely high for every task (in all cases over 97%), showing that BPE can effectively leverage vocabulary differences among domains. The RRR and UNF values were increased with respect to the original BPE segmentation (Table 1). This is unsurprising: as we work with different domains, we now have more n-grams from our test documents which were unseen in the Europarl training data. Moreover, since rare words tend to be split by the BPE process, it is likely to have more words broken up in the test documents with the BPE from Europarl. Therefore, the repetition rate is increased. Finally, it is also remarkable that language pairs involving German usually obtained the highest values of UNF. This is because German is more inflective than English or French. Therefore, it is likely to have higher UNF. Since we applied joint BPE, this also affected to the corresponding part in English.

As we assumed no in-domain data and a potentially unknown domain, we lacked of development sets for this task. Therefore, we used the same algorithm and learning rate for all tasks. We took advantage of the exploration carried out in Section 7.1. Following Table 4, we applied Adadelta with a learning rate of 0.1.

7.2.1. Translation post-editing with online learning

First, we compare the effort required for post-editing the outputs of the neural system. Table 8 shows the results of translation quality, in terms of TER and BLEU, for static and adaptive NMT systems. As expected, the translation quality was much lower than in the previous scenario. Differences were especially severe in the XRCE and EU tasks. This is mostly due to the features of each corpora.

The XRCE task relates to printer manuals and contains many short sentences, referring to technical details. Additionally, such manuals usually have formatting templates. Since the system never faced such templates, it made many mistakes. Note that the TER was extremely high in this task. This phenomenon, also observed by Chinea-Rios et al. (2017) for the same task, is due to the translation of short sentences with a NMT system trained on long sentences from a different domain (Europarl). Therefore, the system generated hypotheses much longer than the references. Therefore, in order to match the reference, TER must delete many words. This problem may be addressed via heuristics in the search method (as pointed out by Chinea-Rios et al. (2017)), but this is out of the scope of this work.

The EU task is also highly structured. As it records the European Union Bulletin, it contains many

sentences from official templates, which have a particular, formal style. It also contains many records with a rigid formatting template, as in the XRCE task. The Europarl corpus mostly records speeches given in the European Parliament. Hence, differences among the EU and the Europarl corpora are notorious.

Table 8: Translation post-editing results, in terms of TER [%] and BLEU [%], for adaptive NMT systems (OL-INMT) compared to static NMT. The NMT system was trained exclusively on Europarl data. ∇ and Δ represent absolute decrements and increments in terms of percentage of the corresponding metric. Results that are statistically significantly better for each task and metric are boldfaced.

		TER [%]			BLEU [%]		
		NMT	OL-NMT	∇	NMT	OL-NMT	Δ
XRCE	En→De	86.2 ± 0.9	$\mathbf{68.3 \pm 1.1}$	17.9	6.6 ± 0.5	$\mathbf{20.4 \pm 0.9}$	13.8
	En→Fr	76.3 ± 1.1	$\mathbf{58.9 \pm 1.0}$	17.4	12.8 ± 0.6	$\mathbf{29.1 \pm 1.0}$	16.3
EU	En→De	73.5 ± 0.9	$\mathbf{69.4 \pm 0.9}$	4.1	18.1 ± 0.6	$\mathbf{20.8 \pm 0.6}$	2.7
	En→Fr	59.6 ± 0.7	$\mathbf{56.5 \pm 0.9}$	3.1	27.7 ± 0.6	$\mathbf{33.2 \pm 0.6}$	5.5
UFAL	En→De	66.9 ± 0.6	$\mathbf{62.8 \pm 0.6}$	4.1	15.7 ± 0.4	$\mathbf{18.8 \pm 0.4}$	3.1
	En→Fr	52.9 ± 0.6	$\mathbf{49.9 \pm 0.6}$	3.0	29.5 ± 0.5	$\mathbf{33.4 \pm 0.5}$	3.9
TED	En→De	61.5 ± 0.5	$\mathbf{56.8 \pm 0.5}$	4.7	20.4 ± 0.4	$\mathbf{23.4 \pm 0.5}$	3.0
	En→Fr	57.3 ± 0.5	$\mathbf{52.7 \pm 0.5}$	4.6	27.3 ± 0.5	$\mathbf{31.1 \pm 0.5}$	3.8

On the other hand, the UFAL and TED corpora are closer to Europarl. Although the domains are different (medical and talks), the style, constructions and template of these documents are similar to Europarl. Therefore, the differences in terms of translation quality are smaller. Fig. 5 shows examples of common sentences from all four tasks.

XRCE	** press " select " to save the setting .* ** press " output " to select " on " .*
EU	*31996 Y 0801 (04) Council Resolution of 15 July 1996 on the transparency of vocational training certificates .* *Reference : Conclusions of the Vienna European Council : Bull. 12-1998 , points I. 19 et seq.*
UFAL	*It 's a long , hollow tube at the end of your digestive tract where your body makes and stores stool .* *We are also studying how their work affects the quality of their lives .*
TED	*Everybody talks about happiness these days .* *I 'm going to talk today about energy and climate .*
Europarl	*A Republican strategy to counter the re-election of Obama* *Republican leaders justified their policy by the need to combat electoral fraud .*

Figure 5: Examples of the XCRE, EU, UFAL and TED tasks, in English. All sentences belong to the corresponding test set. Sentences from the XRCE and EU corpora are highly structured, while the other tasks have a more natural style.

As expected, the development of adaptive NMT systems greatly improved the quality of the systems. The improvements brought by continuous learning to the XRCE task were large (17.9/17.4 points in terms of TER and 13.8/16.3 in terms of BLEU). These large improvements were due to the aforementioned structure features of this text. Since the text was extracted from printer manuals, the restricted repetition rate was extremely high: more than a 25% in al cases (see Table 7).

OL is more effective in texts with high RRR, since upcoming events have been already seen. This effectiveness is boosted by our experimental conditions: we were translating with a general NMT system. Therefore, the systems was prone to make the same errors over and over. As we introduced continuous learning in the system, it rapidly adapted to the XRCE features. Since the XRCE test set is quite repetitive, the OL-NMT avoided to make the same error again, which had a great impact in effort reduction and translation quality.

OL was also effective for the rest of tasks, obtaining consistent improvements, ranging from 3.0 to 4.7 TER points. It is worth noting that for the TED (En→De) task, a system exclusively trained on out-of-domain data and fine-tuned via OL, was able to outperform a PB-SMT system trained on in-domain data

(Table 2). The rest of OL systems also behaved good, achieving performances close to the systems trained on in-domain data. These results demonstrate that online learning is a good choice when developing translation systems with scarce data resources.

This experiment is comparable to the *a posteriori* adaptation strategy developed by Turchi et al. (2017). They also adapted a general model to a given domain by means of incremental learning on post-edited samples, obtaining significant BLEU improvements.

7.2.2. Interactive machine translation with online learning

Next, we study the effectiveness of the general NMT system in the interactive framework and the effect of OL-based adaptation in terms of effort reduction. Table 9 shows the INMT results of an adaptive system and a static one.

Table 9: Human effort required for INMT systems with online learning (OL-INMT), compared to static INMT, in terms of KSMR [%]. NMT systems were exclusively trained on Europarl data. ∇ represents absolute decrements in percentage. Results that are statistically significantly better for each task and metric are boldfaced.

		KSMR [%]		
		INMT	OL-INMT	∇
XRCE	En→De	52.2 ± 0.6	$\mathbf{35.1 \pm 0.7}$	17.1
	En→Fr	56.6 ± 0.7	$\mathbf{32.9 \pm 0.7}$	23.7
EU	En→De	28.0 ± 0.4	$\mathbf{26.4 \pm 0.5}$	1.6
	En→Fr	27.0 ± 0.5	$\mathbf{23.2 \pm 0.4}$	3.8
UFAL	En→De	33.3 ± 0.4	$\mathbf{29.7 \pm 0.3}$	3.6
	En→Fr	29.6 ± 0.3	$\mathbf{25.6 \pm 0.3}$	4.0
TED	En→De	30.2 ± 0.3	$\mathbf{28.0 \pm 0.3}$	2.2
	En→Fr	28.3 ± 0.3	$\mathbf{26.0 \pm 0.3}$	2.3

The performance drop of the general INMT system depended on the task: in tasks with domains close to the general corpus, performances of general INMT systems were close to an in-domain system (e.g. TED). But, if the domain of the test data is far from the general corpus (XRCE), the human effort required dramatically rose.

The introduction of OL into the interactive systems, had a similar effect to that observed in Table 8: we obtained significant KSMR reductions for all tasks. The greatest improvements were again obtained in the XCRE task, due to the aforementioned reasons (highest RRR and shortest sentences). In the case of the TED task, online learning overcame the gap between training a specific system or using the general one. Interestingly, an adaptive INMT system, trained on out-of-domain data, performed better than PB-SMT state-of-the-art systems (Table 3).

7.3. Scenario #3: Fine-tuning a general neural machine translation system

In our last experimental setup, we hybridized scenarios #1 and #2: we have available in-domain and out-of-domain data. Thus, we started from a general NMT system, trained on an out-of-domain corpus, and fine-tuned it with the in-domain training data. Finally, we followed the OL refinement procedure, as in previous scenarios. We study if OL can bring enhancements to an already fine-tuned system, and if so, to what extent.

Again, we used the Europarl corpus as out-of-domain. We followed the same segmentation strategy than in Section 7.2, applying it also to the training set. Table 7 shows the vocabulary coverage of the training sets, generally high. Only in the UFAL corpus the coverage was slightly lower (from 85.9% to 92.5%).

Once we had our system trained on Europarl, we continued the training on each in-domain training set. For this retraining, we kept the hyperparameters used for training the original NMT system: Adam with $\rho = 0.0002$. Following Wu et al. (2016), we also tested vanilla SGD with learning rate annealing, but we

obtained poorer results. We early-stopped the training following the same criterion as in the general case (Section 6.4), but setting the patience to 10 evaluations.

We repeated the same grid search exploration for obtaining the best hyperparameters for each OL optimizer, as in Section 7.1. Interestingly, the top-performing algorithms were the same in both scenarios. Adadelta or SGD obtained the best performance. But in this case, they required higher learning rates. Table 10 shows the best configuration for each task.

Table 10: Best online SGD optimizer for each task for fine-tuned NMT systems.

		Algorithm	ρ
XRCE	En→De	Adadelta	0.5
	En→Fr	Adadelta	0.5
EU	En→De	SGD	10^{-4}
	En→Fr	Adadelta	0.1
UFAL	En→De	Adadelta	0.5
	En→Fr	Adadelta	0.5
TED	En→De	Adadelta	0.1
	En→Fr	SGD	0.01

7.3.1. Translation post-editing with online learning

As in previous sections, we first compare static and adaptive systems, in terms of translation quality and effort required in the post-editing process. Table 11 shows the results of the systems initialized with Europarl and fine-tuned with their corresponding in-domain training data.

Table 11: Translation post-editing results, in terms of TER [%] and BLEU [%], for adaptive NMT systems (OL-INMT) compared to static NMT. All NMT systems have been pretrained on Europarl data and fine-tuned with the training data from each task. ∇ represents absolute decrements in terms of percentage of the corresponding metric. Results that are statistically significantly better for each task and metric are boldfaced.

		TER [%]			BLEU [%]		
		NMT	OL-NMT	∇	NMT	OL-NMT	Δ
XRCE	En→De	60.0 ± 1.0	$\mathbf{53.1 \pm 1.1}$	6.9	27.3 ± 1.2	$\mathbf{33.6 \pm 1.2}$	6.3
	En→Fr	50.1 ± 1.0	$\mathbf{43.2 \pm 1.1}$	6.9	38.8 ± 1.2	$\mathbf{46.6 \pm 1.2}$	7.8
EU	En→De	51.8 ± 1.0	51.7 ± 1.0	0.1	37.1 ± 1.0	36.8 ± 1.1	−0.3
	En→Fr	37.9 ± 0.8	38.2 ± 0.8	0.0	50.6 ± 0.7	51.8 ± 0.9	0.2
UFAL	En→De	56.5 ± 0.6	$\mathbf{53.3 \pm 0.6}$	3.2	23.4 ± 0.6	$\mathbf{26.0 \pm 0.6}$	2.6
	En→Fr	47.8 ± 0.6	$\mathbf{40.9 \pm 0.5}$	6.9	36.6 ± 0.6	$\mathbf{42.8 \pm 0.6}$	6.2
TED	En→De	52.2 ± 0.5	$\mathbf{51.0 \pm 0.5}$	1.2	27.5 ± 0.6	$\mathbf{29.8 \pm 0.6}$	2.3
	En→Fr	47.5 ± 0.5	46.8 ± 0.5	0.7	36.4 ± 0.5	37.0 ± 0.5	0.6

Compared to systems exclusively trained on in-domain data (Table 2), we found that those initialized from Europarl performed generally better if we have scarce in-domain data. In the XRCE and TED tasks, both TER and BLEU were significantly improved when fine-tuning the general system. We found improvements of approximately 4 TER points in each task. On the other hand, if we had available a large amount of in-domain data, the fine-tuning effectiveness is diluted. This is the case of the UFAL task: as this is a large in-domain corpus, to pre-train with Europarl had minor effects on the final NMT performance.

The addition of OL to the NMT systems generally improved the performance with respect to static systems. In the XRCE and UFAL tasks, the improvements were large: From 3.2 to 6.9 TER points. According to their RRR and UNF metrics (Table 7), this was expected, because those are the corpora with higher RRR. The TED task was also benefited from OL, but to a lower extent. Even though, we observed significant improvements in the En→De.

On the other hand, in the EU task we obtained almost no differences between static and adaptive systems. This lack of improvement was a common pattern in the UE along all the experimentation conducted in this work. As stated in Section 7.1.1, this is because the RRR and UNF values of this corpus. OL was unable to be exploited to the full in this corpus.

7.3.2. Interactive machine translation with online learning

Table 12 shows the effort required in a IMT scenario. Compared to the systems exclusively trained on the in-domain data (Table 6), we observed the same phenomenon as in the previous section: the usage of out-of-domain data was especially effective in tasks with scarce in-domain data. Fine-tuned systems performed clearly better in all cases but UFAL (En→De).

Table 12: Effort required for adaptive NMT systems (OL-INMT) compared to static NMT, in terms of KSMR [%]. All NMT systems have been pretrained on Europarl data and fine-tuned with the training data from each task. ∇ represents absolute decrements in terms of percentage. Results that are statistically significantly better for each task and metric are boldfaced.

		KSMR [%]		
		INMT	OL-INMT	∇
XRCE	En→De	23.3 ± 0.6	$\mathbf{20.3 \pm 0.6}$	3.0
	En→Fr	22.4 ± 0.5	$\mathbf{17.8 \pm 0.5}$	4.6
EU	En→De	16.4 ± 0.4	16.3 ± 0.4	0.1
	En→Fr	14.0 ± 0.4	13.6 ± 0.4	0.3
UFAL	En→De	22.9 ± 0.3	$\mathbf{21.7 \pm 0.3}$	1.2
	En→Fr	18.8 ± 0.3	$\mathbf{17.2 \pm 0.3}$	1.6
TED	En→De	23.8 ± 0.3	$\mathbf{23.0 \pm 0.3}$	0.8
	En→Fr	21.8 ± 0.3	$\mathbf{21.2 \pm 0.3}$	0.6

The largest improvements were obtained in the XRCE and TED tasks, with less training data. In these cases, the enhancements ranged from 4.2 to 8.9 KSMR points. The UE task is also improved if we initialize our models with Europarl, obtaining diminishes of 2.3 and 3.4 KSMR points. Finally, fine-tuning had a minor effect on the UFAL task, as the in-domain corpus is large enough to build a good INMT system. Nevertheless, pre-training was harmless and we obtained improvements in the En→Fr language pair. On the other hand, in the En→De language pair, we obtained a minor degradation of the model (+0.4% KSMR points).

The addition of OL to these systems had similar effects than in the previous scenarios. Adaptive systems performed significantly better than static systems in all tasks but EU, due to the aforementioned reasons: the RR, RRR and UNF values.

7.4. Further analyses

Finally, we analyze additional aspects of the proposed systems: the effectiveness of the cLSTM units, response times and computational overhead of the adaptation process via online learning and the effectiveness of the proposed vocabulary-masking strategy. Finally, in order to obtain additional insights of the adaptation via OL in NMT, we show and analyze some qualitative examples.

7.4.1. Evaluating the cLSTM units

We evaluate the performance of the cLSTM unit proposed in Section 3. To that end, we compare the results obtained using regular LSTM units in the decoder, against using cLSTM ones. We trained the same systems than in Section 7.1, but using standard LSTM units (Gers et al., 2000) with attention in the decoder. The rest of hyperparameters of the NMT models remained the same.

The results of this experimentation are shown Table 13, in terms of translation quality (TER and BLEU). In all cases, the systems featuring cLSTM units performed better than those with classical LSTM. The differences were consistent and constant across all tasks: cLSTM units increased the BLEU from around 2 to 3 points. TER was also improved in all cases, in this case from around 2 to almost 5 points.

In addition and despite having more parameters than regular LSTMs, we did not find a significant computational overhead when using cLSTM units, neither in the training and decoding phases. Hence, we conclude that the inclusion of the conditional mechanism to LSTM units was highly positive for the NMT model.

Table 13: Results of translation quality for all tasks in terms of TER [%] and BLEU [%] comparing regular LSTM units and cLSTM units. cLSTM figures are the same than in Table 2. Results that are statistically significantly better for each task and metric are boldfaced.

		TER [%]		BLEU [%]	
		LSTM	cLSTM	LSTM	cLSTM
XRCE	En→De	67.1 ± 1.0	$\mathbf{64.1 \pm 1.1}$	21.5 ± 1.1	$\mathbf{24.4 \pm 1.1}$
	En→Fr	57.1 ± 1.1	$\mathbf{54.0 \pm 1.1}$	32.9 ± 1.2	$\mathbf{34.7 \pm 1.2}$
EU	En→De	57.5 ± 1.0	$\mathbf{54.6 \pm 1.0}$	32.4 ± 1.1	$\mathbf{35.8 \pm 1.1}$
	En→Fr	41.0 ± 0.8	$\mathbf{39.3 \pm 0.8}$	47.4 ± 0.9	$\mathbf{50.0 \pm 0.9}$
UFAL	En→De	58.3 ± 0.6	$\mathbf{56.5 \pm 0.6}$	21.5 ± 0.5	$\mathbf{24.2 \pm 0.5}$
	En→Fr	49.7 ± 0.6	$\mathbf{46.4 \pm 0.6}$	35.2 ± 0.6	$\mathbf{37.2 \pm 0.6}$
Europarl	En→De	67.0 ± 0.3	$\mathbf{63.1 \pm 0.4}$	18.1 ± 0.3	$\mathbf{20.0 \pm 0.3}$
	En→Fr	62.2 ± 0.3	$\mathbf{55.0 \pm 0.3}$	25.3 ± 0.3	$\mathbf{27.8 \pm 0.3}$
TED	En→De	59.2 ± 0.5	$\mathbf{55.5 \pm 0.6}$	22.6 ± 0.5	$\mathbf{24.5 \pm 0.5}$
	En→Fr	54.9 ± 0.5	$\mathbf{51.5 \pm 0.5}$	29.3 ± 0.5	$\mathbf{32.1 \pm 0.5}$

7.4.2. Temporal costs of interactive, adaptive NMT

This work is framed in an interactive protocol, therefore, response times of the system must be adequate for giving the user the feeling of real-time interaction. According to Nielsen (1993), a response time below 0.1 seconds gives the user a feeling of instantaneous reaction. If the response time is between 0.1 and 1 seconds, the user will notice a delay, but his/her flow of though would stay uninterrupted. Moreover, since we apply OL after interactively translating each sample, the retraining time should also be considered.

Table 14 shows the response and learning times for each task1[4]. These values refer to the first scenario (Section 7.1). In the other scenarios we used as NMT system the one trained on Europarl. Therefore, this is the reference for those cases.

Response times were around 0.1 and 0.3 seconds. These values are close to the 0.1 seconds specified by Nielsen (1993) for having a real-time user feeling. Therefore, users would notice a slight delay on system reactions, but their focus during the interactive translation process will hopefully be unaffected. Nevertheless, we should confirm this by means of an experimentation with real users.

Table 14: Average interaction response time (RT) and learning time (LT) for all tasks.

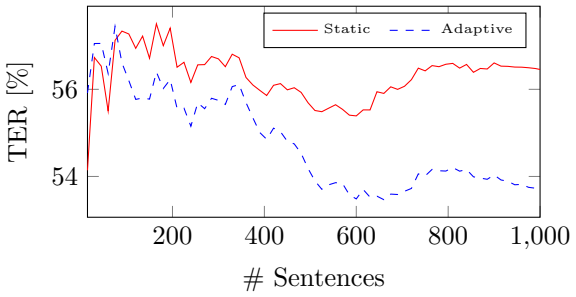
		RT (s)	LT (s)
XRCE	En→De	0.14	0.09
	En→Fr	0.12	0.08
EU	En→De	0.28	0.13
	En→Fr	0.31	0.13
UFAL	En→De	0.26	0.15
	En→Fr	0.26	0.15
Europarl	En→De	0.16	0.14
	En→Fr	0.13	0.14
TED	En→De	0.23	0.12
	En→Fr	0.19	0.12

On the other hand, the learning times were kept constant, regardless the task. Learning times were around 0.1 seconds, therefore, differences between adaptive and static systems were almost unnoticeable in terms of usability.

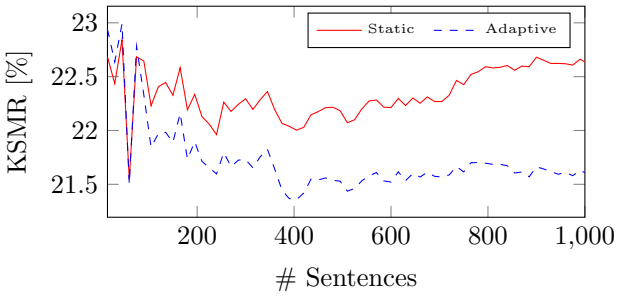
7.4.3. Impact of online learning

We deepen in the effects of continuous learning in NMT, comparing adaptive versus non-adaptive systems in translation post-editing and IMT. Since we want to reduce the effort required by the user, we are interested in TER and KSMR. We measured cumulative TER and KSMR, as the post-editing and IMT processes advanced. We report results (Fig. 6) from the UFAL En→De task, for the systems trained on in-domain and out-of domain data (Section 7.1 and Section 7.2, respectively).

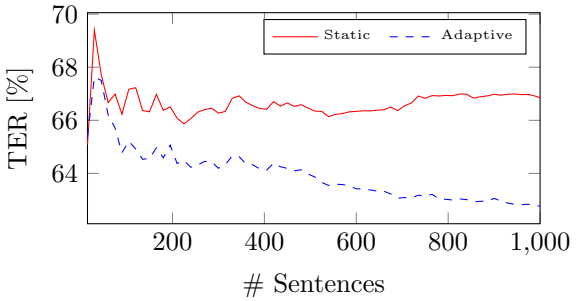
[4]Experiments executed on a single GeForce GTX 1080 GPU.



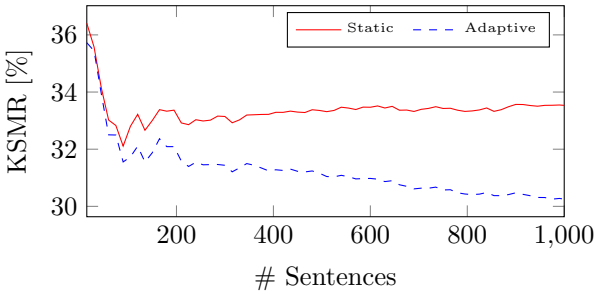
(a) Cumulative TER for UFAL En→De. Systems trained on in-domain data.



(b) Cumulative KSMR for UFAL En→De. Systems trained on in-domain data.



(c) Cumulative TER for UFAL En→De. Systems trained on Europarl.



(d) Cumulative KSMR for UFAL En→De. Systems trained on Europarl.

Figure 6: Cumulative TER and KSMR of static (solid lines) and adaptive (dashed lines) NMT systems for the UFAL En→De task. Plots Fig. 6a and Fig. 6b refer to systems trained on in-domain data, while results of Fig. 6c and Fig. 6d were obtained with a system trained only on out-of-domain data.

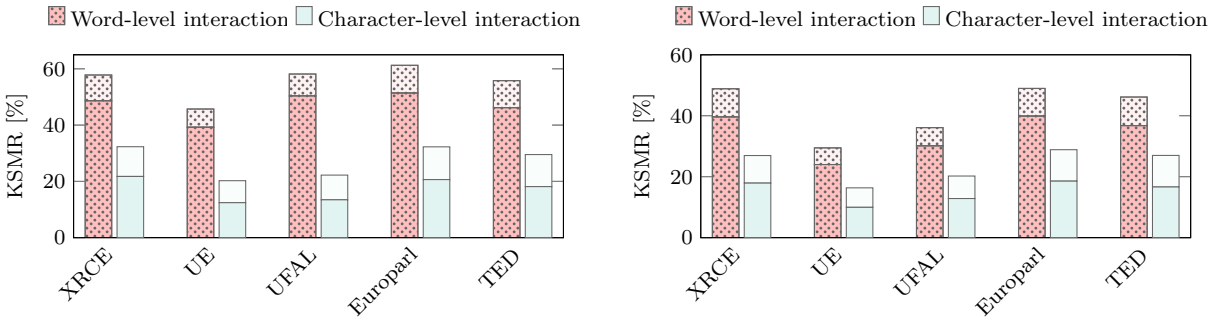
As shown in Fig. 6a and Fig. 6b, the adaptive systems were able to rapidly take advantage of the post-edited samples. With approximately 100 samples, TER and KSMR were considerably lowered; with 600 sentences, the differences were large. From here, the systems get on a performance plateau. Nevertheless, if we attend to the static system, we observe that from the sentence 600, the task becomes more difficult, and the TER and KSMR were increased. OL prevented some of this rise, stabilizing the performance of the systems.

OL applied to systems exclusively trained on out-of-domain data (Fig. 6c and Fig. 6d), improved the performance of the systems. Both TER and KSMR followed a continuous drop. Although expected, this behavior confirms that the systems could be enhanced to larger extents by means of continuous learning, provided that we had more data.

7.4.4. Vocabulary-masking strategy

We introduced (Section 4.1) a simple yet effective way for performing character-level interactions on a NMT system that works at word (or subword) level. A system with character-level interactions will potentially require less keystrokes than another based on word-level interactions, provided that it is able to correctly profit the user feedback. On the other hand, the number of mouse actions may be increased in character-level systems, since the user can move the mouse along one word for correcting it, spending more than one mouse action. In word-based interaction, words are treated as atomic units; therefore, the number of mouse actions required is potentially lower.

In order to assess the proposed character-based INMT systems, we measured the KSMR required by the same INMT system when performing interactions at either word or character level. Results are shown in Fig. 7. The INMT systems are those from Table 3. From KSMR, we differentiated keystrokes and mouse actions.



(a) En→De (b) En→Fr

Figure 7: KSMR of INMT systems of all tasks. We compare word-level interaction (dotted) versus character-level interaction. From each bar, the upper (lighter) part represents the mouse action fraction of KSMR, and the lower part accounts for the keystrokes.

According to Fig. 7, to perform character level interactions greatly diminished the number of keystrokes required. Reductions were around 50% in the case of French and even larger in the case of German (from a 60% to a 75%). This suggests that the system was able to correctly predict even the long and compounded words from German. As expected, character-based interaction slightly rose the amount of mouse actions required. Nevertheless, the increase of mouse actions was small.

Comparing both levels of interactions, conclusions are indisputable: to perform character-level interaction is more effective than the word-level one, in terms of the human effort required. Moreover, character-level interaction allows the user to have a more precise and natural control of the IMT process.

7.4.5. Qualitative analysis

We show an example of a real INMT session, using static and online systems. The NMT system was trained only with in-domain data (scenario #1) and the sentence belongs to the UFAL (En→De) task.

Source (x):		What is the safe and effective route and duration of antibiotic treatment for children with acute pyelonephritis ?
Target ($\hat{y}$):		Was ist die sichere und effektive Methode und Dauer einer Antibiotikabehandlung bei Kindern mit akuter Pyelonephritis ?
IT-0	**MT**	Wie sicher und wirksam und wirkdauer für Kinder mit akuter Pyelonephritis ?
IT-1	**User**	*W*[a]ie sicher und wirksam und Wirkdauer für Kinder mit akuter Pyelonephritis ?
	MT	*Wa*s ist die sichere und wirksame Art und Dauer der Behandlung von Kindern mit akuter Pyelonephritis ?
IT-2	**User**	*Was ist die sichere und* [e]wirksame Art und Dauer der Behandlung von Lindern mit akuter Pyelonephritis ?
	MT	*Was ist die sichere und e*ffektive Art und Dauer der Behandlung von Kindern mit akuter Pyelonephritis ?
IT-3	**User**	*Was ist die sichere und effektive* [M]art und Dauer der Behandlung von Kindern mit akuter Pyelonephritis ?
	MT	*Was ist die sichere und effektive M*enge und Dauer der Antibiotischen Behandlung bei Kindern mit akuter Pyelonephritis ?
IT-4	**User**	*Was ist die sichere und effektive Me*[t]nge und Dauer der Antibiotischen Behandlung bei Kindern mit akuter Pyelonephritis ?
	MT	*Was ist die sichere und effektive Met*hode und Dauer der Antibiotischen Behandlung bei Kindern mit akuter Pyelonephritis ?
IT-5	**User**	*Was ist die sichere und effektive Methode und Dauer* [e]der Antibiotischen Behandlung bei Kindern mit akuter Pyelonephritis ?
	MT	*Was ist die sichere und effektive Methode und Dauer e*iner Antibiotischen Behandlung bei Kindern mit akuter Pyelonephritis ?
IT-6	**User**	*Was ist die sichere und effektive Methode und Dauer einer Antibioti*[k]schen Behandlung bei Kindern mit akuter Pyelonephritis ?
	MT	*Was ist die sichere und effektive Methode und Dauer einer Antibioti*kabehandlung bei Kindern mit akuter Pyelonephritis ?
END	**User**	*Was ist die sichere und effektive Methode und Dauer einer Antibiotikabehandlung bei Kindern mit akuter Pyelonephritis ?*

Figure 8: Real INMT session from the UFAL task (scenario #1). **IT-** refers to the number of iteration of the process, the **MT** row refers to the INMT hypothesis in the current iteration and in the **User** row is shown the feedback introduced by the user: the correct character (boxed). We color in green the prefix that the user has inherently validated while introducing the correction. 12 user actions are required, involving 6 keystrokes and 6 mouse actions (counting final hypothesis acceptation). This represents a KSMR of 10.0%.

The source sentence is "*What is the safe and effective route and duration of antibiotic treatment for*

children with acute pyelonephritis ?" and the desired translation is "*Was ist die sichere und effektive Methode und Dauer einer Antibiotikabehandlung bei Kindern mit akuter Pyelonephritis ?*". The static NMT system proposed the translation "*Wie sicher und wirksam und wirkdauer für Kinder mit akuter Pyelonephritis ?*", which contains several mistakes. Fig. 8 shows the corresponding INMT session for this example. In this case, 6 iterations were required, in order to match the desired translation.

It is interesting to deepen on how the system reacted to the feedback provided. In the first iteration, the user introduced a correction in the interrogative word "*Wie*" ("How"). The system not only correctly predicted the new interrogative word, "*Was*" ("What"), but also changed the hypothesis, matching the correct hypothesis. It is also remarkable that the system naturally handled compound words: in the sixth iteration, the system transformed the words "*Antibiotischen Behandlung*" into the compound "*Antibiotikabehandlung*" from the user keystroke **k**.

Fig. 9 shows the same INMT session, but for an adaptive NMT system. Previously to this sample, the system was already adapted with 915 sentences. The initial hypothesis was better than one proposed by the static system. E.g: the initial interrogative clause was correctly translated. Moreover, it reacted better to the user feedback: in the second iteration, the system correctly predicted the word "*Methode*" with a single keystroke.

Source (x):		What is the safe and effective route and duration of antibiotic treatment for children with acute Pyelonephritis ?
Target ($\hat{y}$):		Was ist die sichere und effektive Methode und Dauer einer Antibiotikabehandlung bei Kindern mit akuter Pyelonephritis ?
IT-0	**MT**	Was ist die sichere und wirksame Art und Dauer der Antibiotischen Behandlung bei Kindern mit akuter Pyelonephritis ?
IT-1	**User**	*Was ist die sichere und* **e**wirksame Art und Dauer der Antibiotischen Behandlung bei Kindern mit akuter pyelonephritis ?
	MT	*Was ist die sichere und e*ffektive Art und Dauer der Antibiotischen Behandlung bei Kindern mit akuter Pyelonephritis ?
IT-2	**User**	*Was ist die sichere und effektive* **M**Art und Dauer der Antibiotischen Behandlung bei Kindern mit akuter pyelonephritis ?
	MT	*Was ist die sichere und effektive M*ethode und Dauer der Antibiotischen Behandlung bei Kindern mit akuter Pyelonephritis ?
IT-3	**User**	*Was ist die sichere und effektive Methode und Dauer* **e**der Antibiotischen Behandlung bei Kindern mit akuter pyelonephritis ?
	MT	*Was ist die sichere und effektive Methode und Dauer e*iner Antibiotischen Behandlung bei Kindern mit akuter Pyelonephritis ?
IT-4	**User**	*Was ist die sichere und effektive Methode und Dauer einer antibioti***k**schen Behandlung bei Kindern mit akuter Pyelonephritis ?
	MT	*Was ist die sichere und effektive Methode und Dauer einer Antibiotik*abehandlung bei Kindern mit akuter Pyelonephritis ?
END	**User**	*Was ist die sichere und effektive Methode und Dauer einer Antibiotikabehandlung bei Kindern mit akuter Pyelonephritis ?*

Figure 9: Same IMT session and notation as Fig. 8 but incorporating OL. Only 4 keystrokes and are 5 mouse actions are now required (KSMR=7.6%).

8. Conclusions and future work

In this work, we studied the application of online learning techniques to NMT. For building our NMT systems, we introduced the cLSTM unit, an extension of the successful cGRU to the LSTM architecture. It consists in the cascaded application of two LSTM blocks in the decoder, with an attention model in between. We presented preliminary results that show that this modification of the LSTM cell offers better performance than standard LSTM units. However, to perform an exhaustive comparison between them is out of the scope of this work and we leave it to future investigations.

We tested our proposals in two computer assisted translation frameworks: post-editing and interactive machine translation. We introduced a novel and effective way for interacting at a character-level with a word-level NMT system. We assessed our strategy and found that it approximately halved the KSMR required during the IMT process.

We studied the application of these systems in three different scenarios. All of them referred to plausible situations in the translation industry and relate to the amount of training data available: we may have enough in-domain data to properly train a NMT system, to also have an out-of-domain corpus to provide additional knowledge to the NMT or we can suffer from lack of in-domain data.

We conducted a wide experimentation, relating two language pairs in five different domains, for each one of the proposed scenarios. The results were conclusive: online learning techniques were able to bring significant improvements to static systems in almost every case. Adaptive NMT systems produced better

translation hypotheses and reduced the human effort required for correcting their outputs. The magnitude of such enhancements were task-dependent, according to the properties of the text to translate. We also faced a discouraging case in which, due to the features of the data, OL was ineffective. We also should investigate solutions to these situations.

The application of online learning to INMT systems reduced even more the human effort required for correcting translation hypotheses. Moreover, the computational overhead of OL was small, making suitable the use of adaptive NMT systems in an interactive scenario. We also compared our system with the state-of-the-art in IMT, based on PB-SMT models. Neural systems beat PB-SMT by a large margin in terms of the human effort required.

As future work, we aim to boost the effectiveness of online learning for NMT. A main issue suffered by SGD is that the objective function (target sentence likelihood) is not necessarily correlated with the assessment criteria (TER, BLEU, KSMR). Therefore, to optimize the likelihood may be suboptimal for the aforementioned metrics. Reinforcement learning has been used for directly optimizing BLEU, as a complementary method to the traditional maximum likelihood training (Wu et al., 2016) or during decoding (Gu et al., 2017). The application of reinforcement learning for adaptive NMT is a promising research direction.

Tightly related to online learning and IMT is the active learning field (Olsson, 2009). Under this paradigm, the system has available a large pool of unlabeled instances and interactively queries to an oracle (e.g. a human agent) to label some of them. It is especially adequate for situations in which the manual labeling is expensive, as machine translation. The active learning framework has brought interesting gains when combined with INMT (Lam et al., 2018; Peris and Casacuberta, 2018a). To further investigate towards this direction seems also encouraging.

Finally, it is mandatory to test our interactive, adaptive systems with real users. We have this point at the top of our agenda and we hope to perform a human evaluation in a near future.

Acknowledgements

The research leading to these results has received funding from the Generalitat Valenciana under grant PROMETEOII/2014/030 and from TIN2015-70924-C2-1-R.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al., 2016. Tensorflow: A system for large-scale machine learning. In: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation. Vol. 16. pp. 265–283.

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., 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., Casacuberta, F., 2011. On multimodal interactive machine translation using speech recognition. In: Proceedings of the International Conference on Multimodal Interaction. pp. 129–136.

Arenas, A. G., 2008. Productivity and quality in the post-editing of outputs from translation memories and machine translation. Localisation Focus 7 (1), 11–21.

Axelrod, A., He, X., Gao, J., 2011. Domain adaptation via pseudo in-domain data selection. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 355–362.

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

Bahdanau, D., Cho, K., Bengio, Y., 2015. Neural machine translation by jointly learning to align and translate, *arXiv:1409.0473*.

Barrachina, S., Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., Lagarda, A., Ney, H., Tomás, J., Vidal, E., Vilar, J.-M., 2009. Statistical approaches to computer-assisted translation. Computational Linguistics 35 (1), 3–28.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J. W., 2010. A theory of learning from different domains. Machine learning 79 (1), 151–175.

Bertoldi, N., Cettolo, M., Federico, M., 2013. Cache-based online adaptation for machine translation enhanced computer assisted translation. Proceedings of the XIV Machine Translation Summit, 35–42.

Bojar, O., Buck, C., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., Kreutzer, J., 2017a. Proceedings of the Second Conference on Machine Translation.

Bojar, O., Graham, Y., Kamran, A., 2017b. Results of the wmt17 metrics shared task. In: Proceedings of the Second Conference on Machine Translation. pp. 489–513.

Bojar, O., Haddow, B., , D. M., Sudarikov, R., Tamchyna, A., Variš, D., 2017c. Report on building translation systems for public health domain (deliverable D1.1). Tech. Rep. H2020-ICT-2014-1-644402, Technical report, Health in my Language (HimL).

Britz, D., Goldie, A., Luong, M.-T., Le, Q., 2017. Massive exploration of neural machine translation architectures. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 1442–1451.

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

Cai, D., Zhang, H., Ye, N., 2013. Improvements in statistical phrase-based interactive machine translation. In: Proceedings of the International Conference on Asian Language Processing. pp. 91–94.

Casacuberta, F., Civera, J., Cubel, E., Lagarda, A. L., Lapalme, G., Macklovitch, E., Vidal, E., 2009. Human interaction for high-quality machine translation. Communications of the ACM 52 (10), 135–138.

Castaño, M.-A., Casacuberta, F., 1997. A connectionist approach to machine translation. In: Proceedings of the International Conference on Theoretical and Methodological Issues in Machine Translation. pp. 160–167.

Chen, B., Cherry, C., Foster, G., Larkin, S., 2017. Cost weighting for neural machine translation domain adaptation. In: Proceedings of the First Workshop on Neural Machine Translation. pp. 40–46.

Cheng, S., Huang, S., Chen, H., Dai, X.-Y., Chen, J., 2016. Primt: A pick-revise framework for interactive machine translation. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1240–1249.

Chiang, D., 2012. Hope and fear for discriminative training of statistical translation models. Journal of Machine Learning Research 13, 1159–1187.

Chinea-Rios, M., Peris, Á., Casacuberta, F., 2017. Adapting neural machine translation with parallel synthetic data. In: Proceedings of the Second Conference on Machine Translation. pp. 138–147.

Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y., 2014. On the properties of neural machine translation: Encoder-decoder approaches. In: Proceedings of the Workshop on Syntax, Semantic and Structure in Statistical Translation. pp. 103–111.

Chung, J., Cho, K., Bengio, Y., 2016. A character-level decoder without explicit segmentation for neural machine translation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. pp. 1693–1703.

Costa-Jussà, M. R., Fonollosa, J. A. R., 2016. Character-based neural machine translation. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics. pp. 357–361.

Crammer, K., Singer, Y., 2001. Ultraconservative online algorithms for multiclass problems. In: Proceedings of the Annual Conference on Computational Learning Theory. pp. 99–115.

Crego, J. M., Kim, J., Klein, G., Rebollo, A., Yang, K., Senellart, J., Akhanov, E., Brunelle, P., Coquard, A., Deng, Y., Enoue, S., Geiss, C., Johanson, J., Khalsa, A., Khiari, R., Ko, B., Kobus, C., Lorieux, J., Martins, L., Nguyen, D., Priori, A., Riccardi, T., Segal, N., Servan, C., Tiquet, C., Wang, B., Yang, J., Zhang, D., Zhou, J., Zoldan, P., 2016. SYSTRAN's pure neural machine translation systems, *arXiv:1610.05540*.

Denkowski, M., Lavie, A., Lacruz, I., Dyer, C., 2014. Real time adaptive machine translation for post-editing with cdec and transcenter. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. pp. 72–77.

Domingo, M., Peris, Á., Casacuberta, F., 2018. Segment-based interactive-predictive machine translation. Machine Translation, 1–23.

Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12, 2121–2159.

Duong, L., Anastasopoulos, A., Chiang, D., Bird, S., Cohn, T., 2016. An attentional model for speech translation without transcription. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 949–959.

Farajian, M. A., Turchi, M., Negri, M., Federico, M., 2017. Multi-domain neural machine translation through unsupervised adaptation. In: Proceedings of the Second Conference on Machine Translation. pp. 127–137.

Federico, M., Bertoldi, N., Cettolo, M., Negri, M., Turchi, M., Trombetti, M., Cattelan, A., Farina, A., Lupinetti, D., Martines, A., Massidda, A., Schwenk, H., Barrault, L., Blain, F., Koehn, P., Buck, C., Germann, U., 2014. The matecat tool. In: Proceedings of the 25th International Conference on Computational Linguistics: System Demonstrations. pp. 129–132.

Forcada, M. L., Ñeco, R. P., 1997. Recursive hetero-associative memories for translation. In: Proceedings of the International Work-Conference on Artificial Neural Networks. pp. 453–462.

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

Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y. N., 2017. Convolutional sequence to sequence learning, *arXiv:1705.03122*.

Gers, F. A., Schmidhuber, J., Cummins, F., 2000. Learning to forget: Continual prediction with LSTM. Neural computation 12 (10), 2451–2471.

González-Rubio, J., Benedí, J.-M., Ortiz-Martínez, D., Casacuberta, F., 2016. Beyond prefix-based interactive translation prediction. In: Proceedings of the Conference on Computational Natural Language Learning. pp. 198–207.

González-Rubio, J., Ortiz-Martínez, D., 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.

Graves, A., 2011. Practical variational inference for neural networks. In: Advances in Neural Information Processing Systems. pp. 2348–2356.

Green, S., Chuang, J., Heer, J., Manning, C. D., 2014. Predictive translation memory: A mixed-initiative system for human language translation. In: Proceedings of the Annual Association for Computing Machinery Symposium on User Interface Software and Technology. pp. 177–187.

Green, S., Heer, J., Manning, C. D., 2013a. The efficacy of human post-editing for language translation. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 439–448.

Green, S., Wang, S., Cer, D., Manning, C. D., 2013b. Fast and adaptive online training of feature-rich translation models. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Vol. 1. pp. 311–321.

Gu, J., Cho, K., Li, V. O., 2017. Trainable greedy decoding for neural machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1958–1968.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9 (8), 1735–1780.

Hokamp, C., Liu, Q., 2017. Lexically constrained decoding for sequence generation using grid beam search. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vol. 1. pp. 1535–1546.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift, *arXiv:1502.03167*.

Jean, S., Cho, K., Memisevic, R., Bengio, Y., 2015. On using very large target vocabulary for neural machine translation. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing. pp. 1–10.

Kaiser, Ł., Nachum, O., Roy, A., Bengio, S., 2017. Learning to remember rare events, *arXiv:1703.03129*.

Kalchbrenner, N., Blunsom, P., 2013. Recurrent continuous translation models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1700–1709.

Kingma, D., Ba, J., 2014. Adam: A method for stochastic optimization, *arXiv:1412.6980*.

Kneser, R., Ney, H., 1995. Improved backing-off for m-gram language modeling. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. Vol. 1. pp. 181–184.

Knowles, R., Koehn, P., 2016. Neural interactive translation prediction. In: Proceedings of the Association for Machine Translation in the Americas. pp. 107–120.

Koehn, P., 2004. Statistical significance tests for machine translation evaluation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 388–395.

Koehn, P., 2005. Europarl: A parallel corpus for statistical machine translation. In: Proceedings of the Machine Translation Summit. pp. 79–86.

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., Herbst, E., 2007. Moses: Open source toolkit for statistical machine translation. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics. pp. 177–180.

Koehn, P., Knowles, R., 2017. Six challenges for neural machine translation. In: Proceedings of the First Workshop on Neural Machine Translation. pp. 28–39.

Koehn, P., Och, F. J., 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. pp. 48–54.

Lagarda, A. L., Ortiz-Martínez, D., Alabau, V., Casacuberta, F., 2015. Translating without in-domain corpus: Machine translation post-editing with online learning techniques. Computer Speech & Language 32 (1), 109–134.

Lam, T. K., Kreutzer, J., Riezler, S., 2018. A reinforcement learning approach to interactive-predictive neural machine translation. In: Proceedings of the European Association for Machine Translation conference. pp. 169–178.

Langlais, P., Lapalme, G., Lorange, M., 2002. TransType: Development-evaluation cycles to boost translator's productivity. Machine Translation 17 (2), 77–98.

Lee, J., Cho, K., Hofmann, T., 2016. Fully character-level neural machine translation without explicit segmentation, *arXiv:1610.03017*.

Libovickỳ, J., Tamchyna, A., Pecina, P., 2016. Adaptation to hungarian, swedish, and spanish (deliverable D1.4). Tech. Rep. H2020-ICT-2014-1-644753, Technical report, KConnect.

Luong, M.-T., Manning, C. D., 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. pp. 1054–1063.

Luong, T., Sutskever, I., Le, Q., Vinyals, O., Zaremba, W., 2015. Addressing the rare word problem in neural machine translation. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing. pp. 11–19.

Macklovitch, E., Nguyen, N.-T., Silva, R., 2005. User evaluation report. Tech. rep., Transtype2 (ISR-2001-32091).

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

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

Mathur, P., Cettolo, M., Federico, M., Kessler, F.-F. B., 2013. Online learning approaches in computer assisted translation. In: Proceedings of the Eighth Workshop on Statistical Machine Translation. pp. 301–308.

Mauro, C., Christian, G., Marcello, F., 2012. Wit3: Web inventory of transcribed and translated talks. In: Conference of European Association for Machine Translation. pp. 261–268.

Murphy, K. P., 2012. Machine learning: a probabilistic perspective. The MIT Press.

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. pp. 160–167.

Och, F. J., Ney, H., 2002. Discriminative training and maximum entropy models for statistical machine translation. In: Pro-

ceedings of the Annual Meeting of the Association for Computational Linguistics. pp. 295–302.
Olsson, F., 2009. A literature survey of active machine learning in the context of natural language processing. Tech. rep., Swedish Institute of Computer Science.
Ortiz-Martínez, D., 2016. Online learning for statistical machine translation. Computational Linguistics 42 (1), 121–161.
Papineni, K., Roukos, S., Ward, T., 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. pp. 311–318.
Pascanu, R., Gulcehre, C., Cho, K., Bengio, Y., 2014. How to construct deep recurrent neural networks, *arXiv:1312.6026*.
Pascanu, R., Mikolov, T., Bengio, Y., 2012. On the difficulty of training recurrent neural networks, *arXiv:1211.5063*.
Pérez-Ortiz, J. A., Torregrosa, D., Forcada, M., 2014. Black-box integration of heterogeneous bilingual resources into an interactive translation system. In: Proceedings of the European Chapter of the Association for Computational Linguistics Workshop on Humans and Computer-assisted Translation. pp. 57–65.
Peris, Á., Bolaños, M., Radeva, P., Casacuberta, F., 2016. Video description using bidirectional recurrent neural networks. In: Proceedings of the International Conference on Artificial Neural Networks. pp. 3–11.
Peris, Á., Casacuberta, F., 2018a. Active learning for interactive neural machine translation of data streams. In: Proceedings of the Conference on Computational Natural Language Learning. pp. 151–160.
Peris, A., Casacuberta, F., 2018b. NMT-Keras: a very flexible toolkit with a focus on interactive NMT and online learning. The Prague Bulletin of Mathematical Linguistics 111, 113–124.
Peris, Á., Cebrián, L., Casacuberta, F., 2017a. Online learning for neural machine translation post-editing, *arXiv:1706.03196*.
Peris, Á., Domingo, M., Casacuberta, F., 2017b. Interactive neural machine translation. Computer Speech & Language 45, 201–220.
Sanchis-Trilles, G., Ortiz-Martínez, D., Civera, J., Casacuberta, F., Vidal, E., Hoang, H., 2008. Improving interactive machine translation via mouse actions. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 485–494.
Schuster, M., Paliwal, K. K., 1997. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45 (11), 2673–2681.
Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Miceli Barone, A. V., Mokry, J., Nadejde, M., 2017. Nematus: a toolkit for neural machine translation. In: Proceedings of the Software Demonstrations at the 15th Conference of the European Chapter of the Association for Computational Linguistics. pp. 65–68.
Sennrich, R., Haddow, B., Birch, A., 2016. Neural machine translation of rare words with subword units. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics. pp. 1715–1725.
Shen, S., Cheng, Y., He, Z., He, W., Wu, H., Sun, M., Liu, Y., 2016. Minimum risk training for neural machine translation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. pp. 1683–1692.
Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J., 2006. A study of translation edit rate with targeted human annotation. In: Proceedings of the Association for Machine Translation in the Americas. pp. 223–231.
Stolcke, A., 2002. SRILM - an extensible language modeling toolkit. In: Proceedings of the International Conference on Spoken Language Processing. pp. 901–904.
Sutskever, I., Vinyals, O., Le, Q. V., 2014. Sequence to sequence learning with neural networks. In: Proceedings of the Advances in Neural Information Processing Systems. Vol. 27. pp. 3104–3112.
Theano Development Team, 2016. Theano: A Python framework for fast computation of mathematical expressions, *arXiv:1605.02688*.
Turchi, M., Negri, M., Farajian, M. A., Federico, M., 2017. Continuous learning from human post-edits for neural machine translation. The Prague Bulletin of Mathematical Linguistics 108 (1), 233–244.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008.
Vinyals, O., Kaiser, Ł., Koo, T., Petrov, S., Sutskever, I., Hinton, G., 2015. Grammar as a foreign language. In: Advances in Neural Information Processing Systems. pp. 2755–2763.
Watanabe, T., Suzuki, J., Tsukada, H., Isozaki, H., 2007. Online large-margin training for statistical machine translation. In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.
Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., Dean, J., 2016. Google's neural machine translation system: Bridging the gap between human and machine translation, *arXiv:1609.08144*.
Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y., 2015. Show, attend and tell: Neural image caption generation with visual attention. In: Proceedings of the International Conference on Machine Learning. pp. 2048–2057.
Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., Courville, A., 2015. Describing videos by exploiting temporal structure. In: Proceedings of the International Conference on Computer Vision. pp. 4507–4515.
Zaidan, O. F., Callison-Burch, C., 2010. Predicting human-targeted translation edit rate via untrained human annotators. In: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 369–372.
Zeiler, M. D., 2012. ADADELTA: An adaptive learning rate method, *arXiv:1212.5701*.