Learning from User Feedback for Machine Translation in Real-Time

Guoping Huang[1,2], Yu Zhou[1], Jiajun Zhang[1], and Chengqing Zong[1]

[1] National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
{guoping.huang,yzhou,jjzhang,cqzong}@nlpr.ia.ac.cn

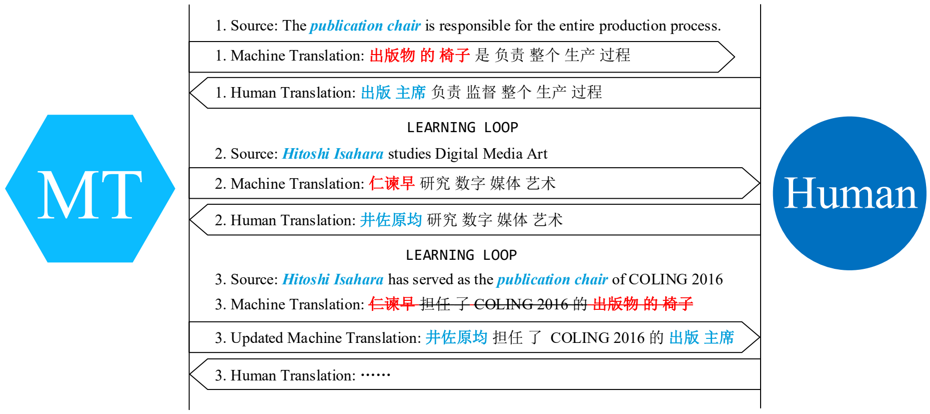
Abstract. Post-editing is the most popular approach to improve accuracy and speed of human translators by applying the machine translation (MT) technology. During the translation process, human translators generate the translation by correcting MT outputs in the post-editing scenario. To avoid repeating the same MT errors, in this paper, we propose an efficient framework to update MT in real-time by learning from user feedback. This framework includes: (1) an anchor-based word alignment model, being specially designed to get correct alignments for unknown words and new translations of known words, for extracting the latest translation knowledge from user feedback; (2) an online translation model, being based on random forests (RFs), updating translation knowledge in real-time for later predictions and having a strong adaptability with temporal noise as well as context changes. The extensive experiments demonstrate that our proposed framework significantly improves translation quality as the number of feedback sentences increasing, and the translation quality is comparable to that of the off-line baseline system with all training data.

1 Introduction

Computer-aided translation (CAT) is a form of language translation in which a human translator uses a software to perform and facilitate the translation process. To further improve the translation efficiency, incorporating the technology of machine translation (MT), especially statistical machine translation (SMT), into the CAT tools has drawn more and more attention. In practice, post-editing is the most popular approach to apply the MT technology to upgrade the CAT system.

In the post-editing scenario, translators generate the translation by correcting the MT results during the translation process. If the raw MT output is good enough, it will take translators little time to achieve the final acceptable translation. Considerable evidence has shown that human translators are more productive and the translation results are more accurate when post-editing is adopted [6, 15, 36, 13]. In the real world, there are a number of CAT tools supporting post-editing, such as SDL Trados and MemoQ.

However, post-editing is far from perfect in fact. So far, MT has focused on providing rough translations for having a glance, rather than outputs that minimize the effort of a translator. As a result, the biggest problem is that the low-quality of MT results often makes a translator disheartened to edit. What's worse, the underlying MT system



The correct key phrase pairs: “publication chair” ||| “出版 主席”; “Hitoshi Isahara” ||| “井佐原均”.

Fig. 1. An overview of learning from users for machine translation in real-time.

will repeat the same errors in the following tasks, in spite of the fact that the translator has corrected them in many times. Therefore, the promotion of machine translation and post-editing is not easy in the human translation community.

Fortunately, the post-editing scenario fits well into an online learning protocol [7], where a stream of human translations is revealed to the MT system one by one as shown in Figure 1. For each source sentence, the system will make an MT output. And then, the automatic translation will be corrected by the translator. As a result, before translating the next sentence, there is a learning loop can be enhanced for the underlying MT system. In the learning loop, the system can use the perfect human translation to upgrade itself to avoid repeating the same translation errors in the following tasks. As we can see, this is an important and challenging problem in the post-editing scenario.

In this paper, to avoid repeating the same MT errors, we present an efficient framework to upgrade the MT system in real-time by learning from user feedback. This proposed framework includes: (1) an anchor-based word alignment model based on Hidden Markov Model (HMM), being specially designed to get correct alignments for unknown words and new translations of known words, for extracting the latest translation knowledge from feedback sentences; (2) an online translation model based on random forests (RFs), updating translation knowledge in real-time for later predictions and having a strong adaptability with temporal noise as well as context changes during translating.

For example, during the English-to-Chinese translation task as shown in Figure 1, the MT results of “publication chair” and “Hitoshi Isahara” are completely wrong. With the help of user feedback, the first step in the learning loop is to successfully grasp the known word “chair” to its new correct translation “主席(chairman)”, and align the unknown words “Hitoshi Isahara” (person name) to the proper Chinese translation “井佐原均”. The key to the solution is integrating phrase segmentation into word alignment under the guidance of alignment anchors, e.g., “responsible ||| 负责” and “studies ||| 研究”. In this paper, we employ mutual information (MI) to find correct alignment anchors with a much higher accuracy rate. As a result, we can get the correct phrase pair candidates: “publication chair ||| 出版主席” and “Hitoshi Isahara ||| 井佐原均”.

The next step is to update the RFs-based online translation model in real-time using the extracted phrase pair candidates with context information. As we can see, the quality of the third sentence has been substantially improved by the updated translation model.

In the experiments, our proposed novel framework significantly improves translation quality with the number feedback sentences increasing, and the translation quality is comparable to that of the off-line baseline system with all training data. So far as we know, it is the first RFs-based translation model.

In summary, this paper makes the following contributions:

(1) The novel anchor-based HMM word alignment model gets more reliable and accurate alignments for unknown words, new translations of known words, and translation knowledge embedded in long sentences. It substantially improves the extraction of the latest translation knowledge from user feedback sentences.
(2) The well designed RFs-based online translation model continuously learns extracted new translation knowledge in real-time. The proposed model significantly outperforms the traditional translation model in terms of online learning.
(3) The proposed online translation model has a strong adaptability with temporal noise and context changes during translating. The algorithm discards trees from forests and continuously grows new trees based on estimated translation errors to promote the translation quality.

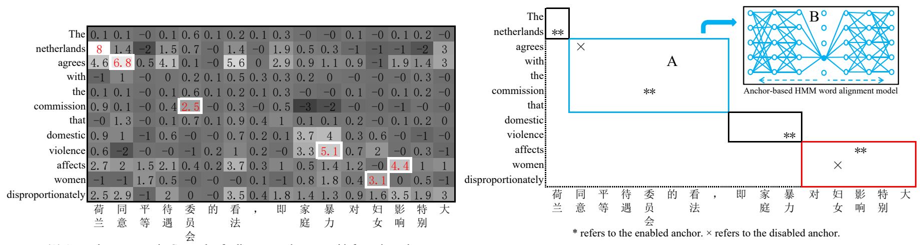
2 Related Work

To update SMT systems in a post-editing scenario where corrected MT output is constantly being returned, previous works can be divided into four types: (1) adapting word alignment model [35, 11, 19, 2, 9], (2) adding new rules to the translation model from the post-edited content [20, 11, 23, 26, 8, 1], (3) updating the target language model [1, 8], and (4) renewing the MT system's discriminative parameters [1, 8].

In this paper, we focus on incremental learning for word alignment model, and online learning for translation model. We first pay attention to alignment problems of unknown words, new translations of known words and long sentences in the post-editing scenario. Second, differing from their work, in this paper, in order to guarantee the comparable performance to the off-line mode, we design the RFs-based translation model, differing from the rule selection model in [12, 18], to implement online learning. What's more, our model can discard old translation knowledge based on estimated translation errors.

3 Online Learning Framework

In this work, we distinguish "online" from "incremental" learning. Online learning has to discard a sample after learning without memorization, and unlike incremental learning allowing to store it. Another related concept is batch learning. If the training data grows, batch learning requires retraining with all previous data and the new data [17]. Our proposed online learning framework includes an anchor-based incremental word alignment model and an online translation model. We will give a detailed description of this framework in next subsections.



(A) A sample sentence pair. Grayscale of cell corresponds to mutual information values.

* refers to the enabled anchor. × refers to the disabled anchor.

(B) one of the segments of the sample sentence pair.

Fig. 2. The anchor-based word alignment model.

3.1 Anchor-based Word Alignment Method

Let $S = s_1^J = s_1 s_2 \dots s_J$ denote the source sentence, and $T = t_1^I = t_1 t_2 \dots t_I$ denote the target sentence, where J and I are the numbers of words in source sentence and target sentence, respectively. Word alignment can be defined as a task to find the optimal sequence $A = a_1 a_2 \dots a_J$, where the expression $a(j) = i$ denotes that the target word t_i is connected to the source word s_j. The standard approach to word alignment makes use of various combinations of five generative models "Model 1-5" [5], HMM-based model [30], and "Model 6" [22]. In machine translation, word alignment plays a crucial role as the precondition.

However, in the learning loop of the post-editing scenario, there are two main challenges for aligning words. The bigger one is the unexpected unknown words and new low frequency translations of known words. That makes it hard for correctly aligning words and further extracting the latest translation knowledge based on only one sentence. For another challenge, long sentences greatly decrease the performance of word alignment. But in the post-editing scenario, we cannot simply filter them as preprocessing in the traditional MT pipeline because of the data sparsity and the long tail theory.

In this paper, we propose a novel anchor-based word alignment method to reduce the scope of alignments of unknown words and new low frequency translations, and meanwhile improve the alignment performance of long sentences. The core idea of the proposed word alignment model is to segment the bilingual sentence pair into bilingual phrases based on alignment anchors before searching the best alignment sequence. In this paper, we employ mutual information (MI) to find alignment anchors with a much higher accuracy rate (over 90%), being similar to [34]:

$$MI[s,t] = \log_2 \frac{P(s,t)}{P(s)P(t)}. \tag{1}$$

This model includes the following three steps as shown in Figure 2:

(1) Find enough anchors based on MI scores as shown in Figure 2(A). (**a**) For each word pair, compute the corresponding MI score according to Equation 1. (**b**) Find the maximum score $MI[s', t']$, and label the cell $[s', t']$ as an anchor. Meanwhile, set $MI[s', \cdot]$ and $MI[\cdot, t']$ to the minimum MI score. (**c**) If the current maximum distance d between adjacent anchors is bigger than the limitation $\mathcal{D}$, repeat Step **b**. We set $\mathcal{D} = 7$ to cooperate the maximum phrase length in translation model.

(2) Segment the sentence pair into bilingual phrases according to the anchors as Zone A of Figure 2(B). Generate *anchor*-centered bilingual phrase set $\{h\}$ with the **restriction**: $\forall\, \overline{d} \in \{h.s.end - anchor.s, h.t.end - anchor.t, anchor.s - h.s.start, anchor.t - h.t.start\}$, $\overline{d} < \mathcal{D}$. To prevent the anchor error to propagate into next steps, each anchor can be disabled with the restriction.

(3) Search the best word alignment using dynamic programming as shown in Zone B of Figure 2(B). In this paper, we employ the modified HMM word alignment model with the anchor constant to align the words of the generated bilingual segments. It should be noted that the initial position of the modified HMM word alignment model is the anchor cell $[anchor.s, anchor.t]$, rather than the random position $[1, \cdot]$ for the original HMM word alignment model. The correct alignment result in Figure 2 is: "The Netherlands {荷兰} agrees {同意} with the commission {委员会} that {，即} domestic {家庭} violence {暴力} affect {影响} women {妇女} disproportionately {特别 大}".

In this paper, the step of incremental word alignment is set to 100 sentences.

3.2 Online Translation Model

3.2.1 RFs-based Translation Model

There has been a recent interest in using random forests (RFs) [4] for natural language processing problems [32]. It has been demonstrated that RFs are better than or at least comparable to other state-of-the-art methods in classification [4, 3]. For MT, RFs provide the following advantages that make them suitable for translation model: (1) decision trees are very fast in both training and classification; (2) they can be easily parallelized, because each tree in a forest is built and test independently from other trees; (3) they are inherently multi-class. In addition, compared to boosting and other ensemble methods, RFs are also more robust against noise [4].

In this paper, we first build up a RFs-based translation model, which combines rich context information for selecting translation rules during decoding. Inspired by [25], the proposed RFs-based translation model is the starting point of online learning from user feedback.

We denote the entire forest for a source phrase as $\mathcal{F} = \{f_1, f_2, \ldots, f_M\}$, where M is the number of decision trees in the forests. Let $f(x, \theta_m) : \mathcal{X} \rightarrow \mathcal{Y}$ denote the m^{th} tree of the forest, where θ_m is a random vector capturing the various stochastic elements of the tree, e.g., the randomly sub-sampled training set and selected random tests at its decision nodes. As a result, given the source phrase $\overline{s}$, the estimated translation probability for the target phrase $\overline{t}$ can be derived as:

$$p(\overline{t}|\overline{s}) = \frac{1}{M} \sum_{m=1}^{M} p_m(\overline{t}|\overline{s}). \tag{2}$$

Each decision tree in a forest is built and tested independently from other trees. During the training, each tree receives a new bootstrapped training set generated by subsampling with replacement of the original training set. We refer to those samples which are not included during the training of a tree as the out-of-bag (OOB) samples, which can be used to compute the out-of-bag-error (OOBE) of the tree. The tests, in form of $g(x) > \theta$, at each decision node of the tree usually contain two parts: (1) a randomly generated test function; (2) a threshold θ which based on the random feature which decides the left/right propagation of samples. The tests are selected by first creating a set of random tests and then picking the best among them according to the entropy:

$$L(\mathcal{R}_j) = -\sum_{\overline{t}=1}^{\overline{T}} p_{\overline{t}}^j \log(p_{\overline{t}}^j) \tag{3}$$

where $p_{\overline{t}}^j$ is the probability of target phrase $\overline{t}$ in node j, and $\overline{T}$ is the number of target phrases.

More specifically, a set of N random tests $\mathcal{S} = \{(g_1(x), \theta_1), \ldots, (g_N(x), \theta_N)\}$ will be created when the node j is created. This node then starts to collect the statistics $p_j = [p_1^j, \ldots, p_{\overline{T}}^j]$ of the samples falling in it. For a random test $d \in \mathcal{D}$, two sets of statistics are also collected: $p_{jld} = [p_1^{jld}, \ldots, p_{\overline{T}}^{jld}]$ and $p_{jrd} = [p_1^{jrd}, \ldots, p_{\overline{T}}^{jrd}]$ corresponding to the statistics of samples falling into left(l) and right(r) partitions according to test d.

The information gain with respect to a test d can be measured as:

$$\varDelta L(\mathcal{R}_j, d) = L(\mathcal{R}_j) - \frac{|\mathcal{R}_{jld}|}{|\mathcal{R}_j|} L(\mathcal{R}_{jld}) - \frac{|\mathcal{R}_{jrd}|}{|\mathcal{R}_j|} L(\mathcal{R}_{jrd}) \tag{4}$$

where $\mathcal{R}_{jld}$ and $\mathcal{R}_{jrd}$ are the left and right partitions made by the test s and $|\cdot|$ denotes the number of samples in a partition. A test with higher gain, produces better splits of the data with respect to reducing the impurity of a node. Therefore, when splitting a node, the test with the highest gain will be chosen as the main decision test of that node.

Draw on the experience of the translation rule selection model introduced by [12, 18], we design the following kinds of features for an extracted phrase pair $\langle s =$"domestic violence", $t =$"家庭 暴力"$\rangle$ from Figure 2 according to [16]:

- **Lexical features**: the 6 words immediately to the left of the source phrase $WS_{s-6} =$ "Netherlands" $, \ldots, WS_{s-1} =$ "that" ; the 6 words immediately to the right of the source phrase $WS_{s+1} =$ "affect" $, \ldots, WS_{s+6} =$ "EOS"; the first word of the source phrase $WSL_s =$ "domestic"; the last word of the target phrase $WSR_s =$ "violence"; the first word immediately to the left of the target phrase $WTL_{t-1} =$ "即"; the last word immediately to the right of the target phrase $WTL_{t+1} =$ "对"; the current lexical weights $P_w(t|s)$ and $P_w(s|t)$; post-editing support $PS = 1$.
- **Length features**: the length of the source phrase $Len_s = 2$ and the length of the target phrase $Len_t = 2$.

We integrate the RFs-based translation model into the log-linear model used by the SMT decoder during the translation of each source sentence. The log-linear model combines features: the translation probabilities $p(t|s)$ and $p(s|t)$ computed by the RFs-based translation model, the lexical weights $p_w(t|s)$ and $p_w(s|t)$, the language model, the reordering model, the word penalty, and the phrase penalty.

Algorithm 1 ORFs-based online translation model

Input: Sequential bilingual phrase pair $\langle s, t\rangle$, the size of the forest T.
Input: The minimum number of samples α, the minimum gain β, the knowledge wighting rate γ.
Output: The forest $\mathcal{F}$.

//For all trees
for $t = 1 \to T$ **do**
  $k \leftarrow$ POISSON(λ)
  if $k > 0$ **then**
    //Update k times.
    for $u = 1 \to k$ **do**
      $j =$ FINDLEAF(s)
      UPDATENODE($j, \langle s, t\rangle$)
      if $|\mathcal{R}_j| > \alpha$ **and** $\exists s \in \mathcal{S}$: $\Delta L(\mathcal{R}_j, s) > \beta$ **then**
        //Find the best test.
        $s_j = \arg\max_{s \in \mathcal{S}} \Delta L(\mathcal{R}_j, s)$
        CREATELEFTCHILD(p_{jls})
        CREATERIGHTCHILD(P_{jrs})
      end if
    end for
  else
    $OOBE_t \leftarrow$ UPDATEOOBE($\langle s, t\rangle$)
    WEIGHTINGTREE($f_t, OOBE_t, \gamma$)
  end if
end for

function WEIGHTINGTREE($f_t, OOBE_t, \gamma$)
  $age_t \leftarrow$ NUMBEROFSAMPLES(f_t)
  if $age_t > \frac{1}{\gamma}$ **and** $OOBE_t >$ RAND( ) **then**
    //Discard the tree.
    $f_t =$ NEWTREE( )
  end if
end function

3.2.2 Online Learning for Translation Model

The original RFs-based translation model is designed to learn in batch or off-line model, i.e., each tree is trained a full sub-set of bilingual phrase pairs. There exist incremental methods for single decision trees but they are either memory intensive, because every node sees and stores all the data [29], or have to discard important information if parent nodes change.

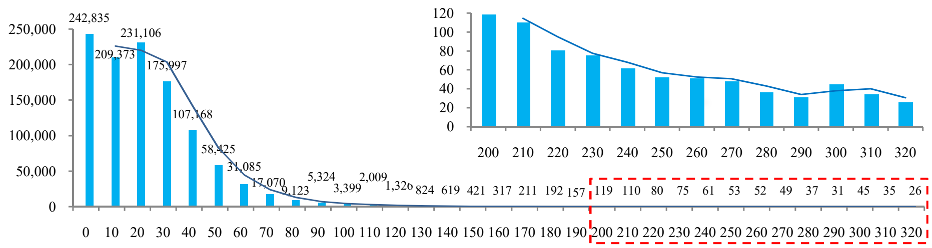
To make the algorithm operate in online learning for translation model, there are two main problems: (1) How to perform bagging in online translation model? (2) How to grow random trees on-the-fly? RFs are ensembles of randomized decision trees combined using bagging. Accordingly, the online version has to combine online bagging [24] and online decision trees with random feature-selection.

(1) Online Bagging

For the bagging part, in this paper, the sequential arrival of the bilingual phrase pairs is modeled by a Poisson distribution. Each tree $f_t(x)$ is updated on each sample k times in a row where k is a random number generated by Poisson(λ) [24].

(2) Online Random Decision Trees

For the growing part, we employ extremely randomized forests [10]: the threshold θ and the test function $g(x)$ of the test $g(x) < \theta$ are chosen *randomly*. During growing of an extremely randomized tree, each decision node randomly creates a set of tests and picks the best according to the test functions and thresholds. When operating in the off-



The horizontal coordinate refers to the sentence length, and the vertical coordinate refers to the number of sentences. For example, the second column (10, 209373) means that there are 209,373 sentences each of which contains 10-19 words.

Fig. 3. A histogram for the length of sentences.

line mode, the decision node has access to all the data falling to that node, and therefore has a more robust estimate of these statistics, compared to node operating. However, in the online mode, the statistics are gathered over time. Therefore, the decision when to split depends on: (1) if there has been enough samples in a node to have a robust statistics; (2) whether the splits are good enough for the classification purpose. Here, we introduce two hyper-parameters: (1) the minimum number of samples α that a node has to see before splitting; (2) the minimum gain a split β that has to achieve before splitting. Thus a node splits if and only if $\mathcal{R}_j > \alpha$ and $\exists d \in \mathcal{D} : \Delta L(\mathcal{R}_j, d) > \beta$.

(3) Temporal Weighting

In post-editing scenario, it requires of the underlying translation model a strong adaptability since the context changes during translating as time goes on. Therefore, we allow our forest to discard the entire tree. To achieve the goal, we can estimate the $OOBE_m$ of the m^{th} tree online. Based on this estimate, we propose to discard trees randomly depending on its OOBE and its age, namely, the number of phrase pairs it has seen so far. By doing this, we can continuously ensure adaptivity throughout time.

In summary, the entire online translation model is depicted in Algorithm 1, where we set $\alpha = 30$, $\beta = 0.1$ and $\gamma = 0.02$.

4 Experiments

We conduct the experiments to test the performance of our proposed framework on improving translation quality with the number of training sentences increasing in the post-editing scenario.

4.1 Experimental Setup

All the experiments are conducted on our in-house developed SMT toolkit including a typical phrase-based decoder and a series of tools, including word alignment and phrase table extraction. All the MT systems are tuned by the development set using ZMERT [33] with the objective to optimize TER [27]. The lower the TER score, the better the translation. And the statistical significance test is performed by the re-sampling approach [14].

We test our method on English-to-Chinese news translation. The training set (1,997,900 sentences, 29,672,190 source words, 27,280,438 target words), development set (1,000 sentences, 27,965 source words, 25,638 target words) and test set (1,100 sentences, 29,570 source words, 26,985 target words) are taken from bilingual news in time order. The histogram for the length of sentences is shown in Figure 3.

Compared to traditional SMT experiments, one big difference is that we preserve the original order of the sentences to simulate post-editing scenario. And additional data employed by this paper includes 10,000,000 Chinese news sentences for training the language model using SRILM toolkit [28]. The maximal entropy based reordering model [31] is trained by the whole training set. As our goal is to test the performance of translation model and word alignment model, we employ the same language model and reordering model for all experiments.

4.2 Results and Analysis

Firstly, we will evaluate the performance of the anchor-based word alignment model and HMM word alignment model on improving the translation quality in the online learning scenario, being denoted as "AnchorAlign" and "HMMAlign", respectively.

Secondly, we evaluate the performance of our RFs- and ORFs-based translation model on improving the translation quality. The baseline system uses precomputed phrase translation probabilities, independent of any other context information. To train the translation model of the baseline system, we run GIZA++ [21] to obtain word alignment in both translation directions, and the word alignment is refined by performing "grow-diag-final" method [16]. The maximum initial phrase length is set to 7. The baseline system, being denoted as "Baseline", contains a traditional off-line translation model. The full training process will be operated each time, including word alignment, phrase extraction, tuning with the development set, and testing with the test set.

Meanwhile, we gather context features based on the word alignment results for training the RFs- and ORFs-based translation models, denoted as "RFs" and "ORFs" respectively. In addition, we have re-implemented two other translation online adaptation methods introduced by [1], i.e., translation adaptation with an external cache, being denoted as "ExternalCache", and an internal cache, being denoted as "InternalCache". Because the principal topic is online learning in this paper, we do not compare among various discriminative translation models or rule selection models.

To improve the clarity, we report all experimental results with respect to the ratio of training sentences in Figure 4. In Figure 4, "RFs" refers to off-line training with respect to the ratio of training sentences. And "ORFs" refers to the online training sentence by sentence, but we reports the results on the test set corresponding to the above ratio.

As shown in Figure 4, all methods will get better results as the training data increasing. Though the cache-based online adaption methods (the dotted lines without a symbol) cannot compare with the off-line baseline system (black solid line without a symbol), it is very simple and easy to implement.

(1) Word Alignment Model: (**A**) If we fix the translation model, such as "ORFs" (■), in terms of TER scores, the performance of the anchor-based word alignment model (the solid lines with symbols, "AnchorAlign") is significantly better than that

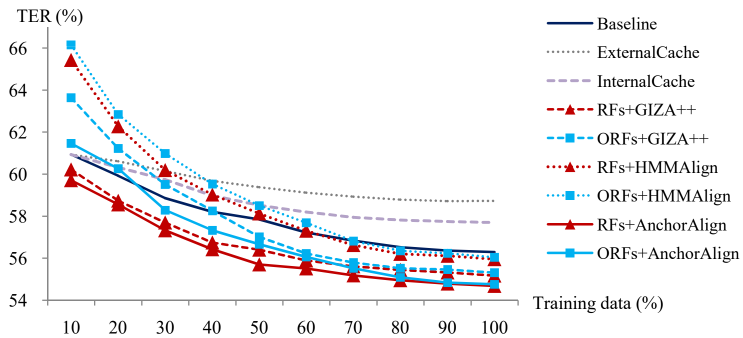


Fig. 4. TER scores with respect to the ratio of training data.

of GIZA++ (the dashed lines with symbols, "GIZA++"), namely reducing 0.55 absolute TER scores, despite of the weak HMM-based word alignment model (the dotted lines with symbols, "HMMAlign"). It means that the anchor-based word alignment model successfully improves the performance by segmenting sentences and achieves better translation results. (**B**) If we focus on the anchor-based word alignment, we can find that it is more suitable for ORF-s based translation model, namely reducing 0.49 absolute TER scores compared to GIZA++. If we were only looking at the TER scores, the improvement is less pronounced. It should be noted that GIZA++ is a combination of various of alignment models (including the HMM word alignment model) and optimization techniques, while the anchor-based alignment model only use the HMM word alignment model. In this perspective, the improvement of the anchor-based word alignment model is remarkable.

(2) Translation Model: In Figure 4, the TER scores of "RFs" and "ORFs" are significantly better than that of other approaches (about 1.0 absolute TER scores). If we fix the word alignment method, the overall performance of the RFs-based translation model is superior to the traditional off-line translation model labeled with "Baseline". The results lay a good foundation for further development of online learning methods. As a result, if we focus on the blue (■) and red (▲) lines, we can find that the performance of the ORFs-based model are comparable with the RFs-based off-line learning model as the number of feedback parallel sentences increasing, and better than that of the traditional translation model (more than 0.9 absolute TER scores). The gap between RFs- and ORFs-based translation model is very small (less than 0.2 TER scores) and can be ignored in practical applications. This means that we have achieved our goals of online learning from user feedback in real-time.

In summary, we can draw the conclusion that the proposed online framework significantly improves the performance of MT outputs by learning from user feedback as the number of training sentences increasing in the post-editing scenario.

5 Conclusion

In this paper, we have presented an efficient framework for updating machine translation by learning from user feedback in real-time. This framework includes an online translation model based on the random forests, and an anchor-based word alignment model which combines phrase segmentation and HMM-based word alignment. It avoids repeating the same MT errors and significantly improves the translation quality as the number of feedback sentences increasing. The experimental results are promising.

Acknowledgements

The research work has been funded by the Natural Science Foundation of China under Grant No. 61303181.

References

1. Bertoldi, N., Simianer, P., Cettolo, M., Wschle, K., Federico, M., Riezler, S.: Online adaptation to post-edits for phrase-based statistical machine translation. Machine Translation 28(3-4), 309–339 (2014)
2. Blain, F., Schwenk, H., Senellart, J.: Incremental adaptation using translation information and post-editing analysis. In: International Workshop on Spoken Language Translation. pp. 234–241 (2012)
3. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM international conference on Image and video retrieval. pp. 401–408 (2007)
4. Breiman, L.: Random forests. Machine learning 45(1), 5–32 (2001)
5. Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics 19(2), 263–311 (1993)
6. Carl, M., Dragsted, B., Elming, J., Hardt, D., Jakobsen, A.L.: The process of post-editing: a pilot study. In: Proceedings of the 8th international NLPSC workshop. Special theme: Human-machine interaction in translation. vol. 41, pp. 131–142 (2011)
7. Cesa-Bianchi, N., Lugosi, G.: Prediction, learning, and games. Cambridge university press (2006)
8. Denkowski, M., Dyer, C., Lavie, A.: Learning from post-editing: Online model adaptation for statistical machine translation. In: Proceedings of ACL 2014 (2014)
9. Farajian, A.M., Bertoldi, N., Federico, M.: Online word alignment for online adaptive machine translation. In: Proceeding of EACL 2014 Workshop on Humans and Computer-Assisted Translation. pp. 84–92 (2014)
10. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine learning 63(1), 3–42 (2006)
11. Hardt, D., Elming, J.: Incremental re-training for post-editing smt. 9th Conference of the Association for Machine Translation in the Americas (AMTA) (2002)
12. He, Z., Liu, Q., Lin, S.: Improving statistical machine translation using lexicalized rule selection. In: Proceddings of COLING 2008. pp. 321–328 (2008)
13. Huang, G., Zhang, J., Zhou, Y., Zong, C.: A new input method for human translators: Integrating machine translation effectively and imperceptibly. In: Proceedings of the IJCAI 2015 (2015)
14. Koehn, P.: Statistical significance tests for machine translation evaluation. In: Proceedings of EMNLP 2004 (2004)
15. Koehn, P.: Computer-added trasnlation. Machine Translation Marathon (2012)
16. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proceedings of HLT-NAACL 2003. pp. 48–54 (2003)

17. Li, L.J., Fei-Fei, L.: Optimol: automatic online picture collection via incremental model learning. International journal of computer vision 88(2), 147–168 (2010)
18. Liu, Q., He, Z., Liu, Y., Lin, S.: Maximum entropy based rule selection model for syntax-based statistical machine translation. In: Proceedings of EMNLP 2008. pp. 89–97 (2008)
19. Mccarley, J.S., Ittycheriah, A., Roukos, S., Xiang, B., Xu, J.M.: A correction model for word alignments. In: Proceedings of EMNLP 2011. pp. 889–898 (2011)
20. Nepveu, L., Lapalme, G., Langlais, P., Foster, G.F.: Adaptive language and translation models for interactive machine translation. In: Proceedings of EMNLP 2004. pp. 190–197 (2004)
21. Och, F.J., Ney, H.: Improved statistical alignment models. In: Proceedings of ACL 2000. pp. 440–447 (2000)
22. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29(1), 19–51 (2003)
23. Ortiz-Martínez, D., García-Varea, I., Casacuberta, F.: Online learning for interactive statistical machine translation. In: Proceedings of NAACL 2010. pp. 546–554 (2010)
24. Oza, N.C.: Online bagging and boosting. In: IEEE International Conference on Systems, Man and Cybernetics. vol. 3, pp. 2340 – 2345 (2005)
25. Saffari, A., Leistner, C., Santner, J., Godec, M., Bischof, H.: On-line random forests. In: IEEE International Conference on Computer Vision Workshops (2009)
26. Simard, M., Foster, G.: Pepr: Post-edit propagation using phrase-based statistical machine translation. Proceedings of the XIV Machine Translation Summit 2013 pp. 191–198 (2013)
27. Snover, M., JDorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Conference of the Association for Machine Translation in the Americas (2006)
28. Stolcke, A., et al.: Srilm - an extensible language modeling toolkit. In: Proceedings of the International Conference on Spoken Language Processing. vol. 2, pp. 901–904 (2002)
29. Utgoff, P.E., Berkman, N.C., Clouse, J.A.: Decision tree induction based on efficient tree restructuring. Machine Learning 29(1), 5–44 (1997)
30. Vogel, S., Ney, H., Tillmann, C.: Hmm-based word alignment in statistical translation. In: Proceedings of the 16th conference on Computational linguistics. vol. 2, pp. 836–841 (1996)
31. Xiong, D., Liu, Q., Lin, S.: Maximum entropy based phrase reordering model for statistical machine translation. In: proceedings of COLING-ACL 2006 (2006)
32. Xu, P., Jelinek, F.: Random forests in language modeling. In: Proceedings of EMNLP 2004. vol. 4, pp. 325–332 (2004)
33. Zaidan: Z-mert: A fully configurable open source tool for minimum error rate training of machine translation systems. The Prague Bulletin of Mathematical Linguistics 91, 79–88 (2009)
34. Zhang, Y., Vogel, S., Waibel, A.: Integrated phrase segmentation and alignment model for statistical machine translation. In: Proceedings of NLP-KE 2003 (2003)
35. Zhao, B., Vogel, S.: Adaptive parallel sentences mining from web bilingual news collection. In: Proceedings of IEEE International Conference on Data Mining 2002. pp. 745–745 (2002)
36. Zhechev, V.: Machine translation infrastructure and post-editing performance at autodesk. In: AMTA 2012 Workshop on Post-Editing Technology and Practice (WPTP 2012). pp. 87–96 (2012)