Using machine translation in computer-aided translation to suggest the target-side words to change

Miquel Esplà-Gomis and **Felipe Sánchez-Martínez** and **Mikel L. Forcada**
Dep. de Llenguatges i Sistemes Informàtics
Universitat d'Alacant, E-03071 Alacant, Spain
{mespla,fsanchez,mlf}@dlsi.ua.es

Abstract

This paper explores the use of machine translation (MT) to help users of computer-aided translation systems based on translation memory to identify the target words in the translation proposals that need to be changed or kept unedited. MT is used as a *black box* to obtain a set of features for each target word in the translation proposals and then used by a binary classifier to determine the target words to change or keep unedited. Experiments conducted in the translation of Spanish texts into English with different corpora shows an accuracy above 96% for fuzzy-match scores above 70%.

1 Introduction

Computer-aided translation (CAT) systems based on translation memory (TM) (Somers, 2003) are the translation technology of choice for most professional translators, especially when translation tasks are very repetitive and effective recycling of previous translations is feasible. The conceptual simplicity of fuzzy-match scores (FMS) (Sikes, 2007) and the ease with which they may be used to determine the confidence on TM proposals are behind this choice.

When using a TM-based CAT system to translate a source segment s', the system provides the set of translation units (TUs) $\{(s_i, t_i)\}_{i=1}^{N}$ whose FMS is above a given threshold Θ. The FMS function measures the similarity between s' and s_i. Even though other FMS functions may be used, we chose for this work one based on the edit distance:

$$\text{score}(s', s_i) = 1 - \text{D}(s', s_i) / \left(\max(|s'|, |s_i|)\right)$$

where $|x|$ is the length (in words) of string x and $\text{D}(x, y)$ refers to the word-based Levenshtein (1966) distance (edit distance) between x and y.

When showing the user a candidate TU, most TM-based CAT tools highlight the words in s_i that differ from those in s' to ease the task of post-editing. It is however up to the translator to identify which target words in t_i should be changed to convert t_i into t', an adequate translation of s'. The method we propose and evaluate in this paper is aimed at automatically recommending which words of t_i should be changed or kept unedited by the translator. From now on, we will refer to this as *word-keeping recommendation*.

To determine the target words that should be changed or kept unedited, source language (SL) and target language (TL) segments are segmented into overlapping sub-segments of variable length and machine-translated into the TL and the SL, respectively. These translations are then used to compute a set of features that are used by a binary classifier to determine which target words in t_i are to be kept unedited and which ones should be changed. The basic idea behind this method is that a word in t_i is likely to be kept if it appears in the translations sub-segments common to s_i and s'. For this task we propose the use of a parametric classifier whose parameters, the set of feature weights, can be obtained in advance from a separate training TM and then used to translate texts from a different domain without a significant loss of accuracy, as demonstrated by our experiments. CAT users could therefore use this MT-based approach in their desktop workstations provided that they have the classifier, on-line access to the MT system(s), and a set of suitable feature weights.

Related work. In the literature one can find several approaches that combine the benefits of MT and TMs beyond the obvious β-*combination* scenario defined by Simard and Isabelle (2009), in which MT is used to translate a new segment when no matching TU above a FMS threshold β is found in the TM. Biçici and Dymetman (2008) integrate a *phrase*-based statistical MT (PBSMT) (Koehn, 2010) system using discontinuous bilingual sub-segments into a TM-based CAT tool. The PBSMT system is trained on the same TM and, when a new source segment s' is to be translated, the segments s_i and t_i in the best matching TU are used to bias the statistical translation of s' towards t_i. This is done by augmenting the PBSMT translation table with bilingual sub-segments coming from the fuzzy match (s_i, t_i). Simard and Isabelle (2009) propose a similar approach in which a new feature function is introduced in the log-linear model combination of a PBSMT system to promote the use of the bilingual sub-segments from the fuzzy match (s_i, t_i). Neither of these two approaches guarantees that the PBSMT system will produce a translated segment containing the translation of the sub-segments that are common to s_i and s'. In contrast, Zhechev and van Genabith (2010) and Koehn and Senellart (2010), who also use a PBSMT system, guarantee that the sub-segments of t_i that have been detected to be aligned with the sub-segments in s_i matched by s' appear in the translated segment.

Our approach differs from those described above in two ways. First, while they use the TM to improve the results of MT, or MT to translate sub-segments of the TUs, our MT-based approach uses MT to improve the experience of using a TM-based CAT system *without actually translating any new material*. Second, the approaches above focus on a specific MT system or family of MT systems (namely, SMT), whereas our MT-based approach uses MT as a black box, and is therefore able to use one or more MT systems at once. In addition, as our MT-based approach does not need to have access to the inner workings of the MT systems, it is capable of using on-line MT systems (thus avoiding any local installation) or even any other source of bilingual information such as dictionaries, glossaries, or terminology data bases.

As regards commercial TM-based CAT tools, *DéjàVu*[1] integrates example-based MT (EBMT) to suggest candidate translations in those cases in which an exact match is not found, but partial matches are available (Lagoudaki, 2008). The EBMT-inspired system is used to propose a translation by putting together sub-segments of the partial matchings available. Unfortunately, we have been unable to find further details on how this method works.

More similar to ours are the work by Kranias and Samiotou (2004), based on the ESTeam CAT system, and the one by Esplà et al. (2011). Kranias and Samiotou (2004) align the words in each TU at different sub-sentential levels by using a bilingual dictionary (Meyers et al., 1998). Then, when a TU is proposed to the CAT user, the alignments previously computed and the MT system are used, respectively, to detect the target words that need to be changed, and to propose a translation for them.

Esplà et al. (2011) use statistical word-alignment (SWA) models computed by means of GIZA++ (Och and Ney, 2003) to align the SL and TL segments of each TU in the TM. Then, when a TU (s_i, t_i) is proposed to the user, the pre-computed word alignments are used to determine the target words to change or keep unedited by computing the likelihood of each word w_{ij} in t_i being kept unedited:

$$p_K^s(w_{ij}, s', s_i, t_i) = \frac{\sum_{v_{il} \in \text{aligned}(w_{ij})} \text{matched}(v_{il})}{|\text{aligned}(w_{ij})|}$$

where $\text{aligned}(w_{ij})$ is the set of source words in s_i that are aligned with the target word w_{ij}, and $\text{matched}(v_{il})$ equals 1 if the source word v_{il} is part of the match between s_i and s', the segment to be translated, and 0 otherwise. This likelihood is then used to decide if w_{ij} is to be changed or kept unedited. It is worth noting that if a word w_{ij} is not aligned to any word in s_i, p_K^s cannot be computed and consequently no recommendation can be made.

In the experiments reported in this paper we compare the performance of our MT-based word-keeping recommendation approach to the SWA-based approach defined by Esplà et al. (2011) and find out that the accuracy of both approaches is quite similar when translating in-domain texts, whereas for out-of-domain texts our MT-based approach achieves higher accuracy. Moreover, in both cases (in-domain and out-of-domain) our MT-based approach has a coverage of 100%, whereas the use of SWA causes the coverage to drop off from around 95% for in-domain texts to 90% for out-of-domain texts.

The rest of the paper is organized as follows. Section 2 presents the features used for word-keeping

[1] http://www.atril.com

recommendation in a binary classification framework. Section 3 describes the binary classifier we have used in our experiments and how it is trained. Section 4 describes the experimental framework, whereas Section 5 discusses the results achieved. The paper ends with some concluding remarks and plans for future work.

2 Word-keeping recommendation as binary classification

When a TU (s_i, t_i) is proposed by the CAT system to the user, a set of features are obtained for each word w_{ij} in t_i, and a binary classifier is then used to determine which words should be kept unedited and which should be changed when modifying t_i in order to get t'. The features we propose to use in this work are based on the assumption that MT can provide evidence about whether each word w_{ij} in t_i should be changed or kept unedited. Let σ be a sub-segment of s_i from one of the matching TUs (s_i, t_i), which is *related by MT* to a sub-segment τ of t_i; by *related by MT* we mean either that machine translating σ leads to τ or vice versa. We hypothesize that:

- if σ is a common sub-segment of both the new segment to translate s' and the source segment s_i, then it is likely that the words in τ will not have to be changed (positive evidence);
- if σ is a sub-segment of s_i but not of s', it is then likely that one or more words in τ will have to be changed (negative evidence).

For example, if a Spanish–English TM contains the pair $(s_i, t_i) =$

("la situación humanitaria parece ser difícil", "the humanitarian situation appears to be difficult"),

the new segment to translate s' is

"la situación política parece ser difícil",

and the MT system provides the following sub-segment pairs (σ, τ) matching (s_i, t_i):

("la","the"), ("situación","situation")*, ("humanitaria", "humanitarian"), ("ser","be")*, ("difícil","difficult")*, ("situación humanitaria","humanitarian situation"), ("ser difícil", "be difficult")*, ("la situación humanitaria", "the humanitarian situation")*

those pairs that also match s' (marked with an asterisk) provide evidence that the words to be kept unedited may be *the*, *situation*, *be* and *difficult*, which is compatible with a possible translation t'="*the political situation appears to be difficult*". Conversely, those pairs (σ, τ) not matching s' provide evidence in favor of changing words *the*, *humanitarian*, and *situation*. Note that, on the one hand, there is some contradiction between this negative evidence and the positive evidence obtained for some words such as *situation*; this will be handled by the classifier by assigning different weights to each feature (see next section). On the other hand, there are some words about which no evidence can be obtained (*appears* and *to*) because they were not matched by the MT system (which, for instance, returned *seems* instead of *appears*).

2.1 Features

From these two types of evidence, we define the features used for word-keeping recommendation. For the word in the j-th position of t_i, w_{ij}, and for each possible source sub-segment length n, we compute the following *positive source* (PS) feature:

$$\mathrm{PS}_n(w_{ij}, s', s_i, t_i) = \\ = \frac{\mathrm{cover}(\mathrm{seg}_n(s_i) \cap \mathrm{seg}_n(s'), \mathrm{seg}(t_i), w_{ij})}{\mathrm{cover}(\mathrm{seg}_n(s_i), \mathrm{seg}(t_i), w_{ij})},$$

where $\mathrm{seg}_n(x)$ stands for the set of length-n sub-segments of x and $\mathrm{cover}(S, T, w_{ij})$ is defined as:

$$\mathrm{cover}(S, T, w_{ij}) = \\ |\{\tau \in T : \exists \sigma \in S \wedge (\sigma, \tau) \in M \wedge w_{ij} \text{ in } \tau\}|,$$

where M is a collection of sub-segment pairs (σ, τ) with $\sigma \in S$ and $\tau \in T$ which are related by MT; that is, $\mathrm{cover}(S, T, w_{ij})$ is the number of target sub-segments $\tau \in T$ containing the word w_{ij} that are related by MT to a sub-segment $\sigma \in S$. A *positive target* (PT) feature is similarly computed for target segments of length n:

$$\mathrm{PT}_n(w_{ij}, s', s_i, t_i) = \\ \frac{\mathrm{cover}(\mathrm{seg}(s_i) \cap \mathrm{seg}(s'), \mathrm{seg}_n(t_i), w_{ij})}{\mathrm{cover}(\mathrm{seg}(s_i), \mathrm{seg}_n(t_i), w_{ij})}.$$

Conversely, analogous negative evidence expressions are used to define negative source (NS) and negative target (NT) features for source and target sub-segments of length n:

$$\mathrm{NS}_n(w_{ij}, s', s_i, t_i) = \\ \frac{\mathrm{cover}(\mathrm{seg}_n(s_i) - \mathrm{seg}_n(s'), \mathrm{seg}(t_i), w_{ij})}{\mathrm{cover}(\mathrm{seg}_n(s_i), \mathrm{seg}(t_i), w_{ij})},$$

$$\mathrm{NT}_n(w_{ij}, s', s_i, t_i) = \frac{\mathrm{cover}(\mathrm{seg}(s_i) - \mathrm{seg}(s'), \mathrm{seg}_n(t_i), w_{ij})}{\mathrm{cover}(\mathrm{seg}(s_i), \mathrm{seg}_n(t_i), w_{ij})}.$$

All these features take values in $[0, 1]$, and may be taken to represent the likelihood, as informed by size-n sub-segments, that word w_{ij} should be kept unedited or changed when modifying t_i to produce t', a valid translation of s'. When both the numerator and the denominator happen to be zero because no matches occur, the value of the feature will be set to 0.5. The FMS is introduced as an additional feature in order to give more weight to predictions coming from close matches and less weight to those coming from fuzzier matches. Following the example above, a recommendation can be given for words *appears* and *to* based on this feature.

3 Parametric classifiers for word-keeping recommendation

It is possible to use any type of binary classifier for word-keeping recommendation. We decided to use a parametric classifier because in this way TM-based CAT users can easily re-use the learned parameters (a vector of feature weights) in different translation tasks and with different TMs. In the experiments we report the results obtained with a perceptron classifier (Duda et al., 2000, Sect. 6.8).

3.1 Perceptron classifier

A perceptron classifier can be used to linearly combine the features for each word, and then to obtain keeping probabilities $p_K(w_{ij}, s', s_i, t_i)$ for each word w_{ij} in t_i by using a sigmoid function:

$$p_K(w_{ij}, s', s_i, t_i) = (1 + e^{-g_k(w_{ij}, s', s_i, t_i)})^{-1} \quad (1)$$

with

$$g_k(w_{ij}, s', s_i, t_i) = \lambda_0 + \sum_{k=1}^{N_F} \lambda_k h_k(w_{ij}, s', s_i, t_i),$$

where N_F is the number of features involved, $h_k(w_{ij}, s', s_i, t_i)$ is the k-th feature, and λ_k is the associated weight. The additional weight λ_0 corresponds to the bias of the perceptron.

3.2 Training

Given a FMS threshold Θ and a training TM —not necessarily the user's TM— training examples are obtained in a leaving-one-out fashion. The method consists of iteratively extracting a TU (s'_m, t'_m) from the training TM and taking its source segment to be a new segment to translate. Then, all the matching TUs $\{(s_i, t_i)\}_{i=1}^{N}$ whose FMS is above the given threshold are extracted from the TM and, for each word w_{ij} in their target segments t_i, the corresponding vector of features is obtained (see Section 2.1). Each vector is then tagged as "keep" or "change", using the function:

$$\hat{p}_K(w_{ij}, s'_m, s_i, t_i) = \begin{cases} 1 & \text{if } w_{ij} \text{ is to be kept} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

which is computed using t'_m as a reference and obtaining the optimal edit-distance path between t_i and t'_m. Once all the examples have been obtained from each segment t_i, the TU (s'_m, t'_m) is reinserted in the TM and the next TU is extracted. This process is repeated until all TUs have been processed.

Training examples obtained in this way are then used to train the classifier above by fitting eq. (1) so as to minimize:

$$E = \sum_{m=1}^{N} \sum_{i \in \mathrm{match}(m,\Theta)} \sum_{j=1}^{|t_i|} E_{ijm} \quad (3)$$

where N is the number of sentence pairs in the training TM, $\mathrm{match}(m, \Theta)$ is the set of indexes of TUs in the TM for which the FMS between them and s'_m is equal or higher than Θ, and

$$E_{ijm} = \mathcal{L}_q(p_K(w_{ij}, s'_m, s_i, t_i), \hat{p}_K(w_{ij}, s'_m, s_i, t_i))$$

is the result of applying the quadratic loss function

$$\mathcal{L}_q(p, \hat{p}) = \frac{1}{2}(p - \hat{p})^2. \quad (4)$$

Since eq. (4) can be differentiated, we have used a gradient descent algorithm (Duda et al., 2000, Section 5.4.2) for optimization.

4 Experimental settings

We have tested our MT-based approach in the translation of Spanish texts into English by using two independent TMs for training, and a test set consisting of two TMs from a common domain. The two TMs used for training are: an in-domain $\mathrm{TM}_{\mathrm{in}}$, from the same domain as the test set, and an out-of-domain $\mathrm{TM}_{\mathrm{out}}$. We choose to have a fixed test set in order to

directly compare the results obtained when training on the in-domain $\mathrm{TM_{in}}$ and on the out-of-domain $\mathrm{TM_{out}}$. In this way, we have been able to test how independent our MT-based method is from the TM used for training. This is a key point, since domain independence will allow CAT users to reuse the feature weights obtained for the classifier without having to run any new training procedure and without having to access other TMs.

The two TMs in the test set are $\mathrm{TM_{trans}}$ and $\mathrm{TM_{test}}$, both from the same domain as $\mathrm{TM_{in}}$. Evaluation was carried out by simulating a CAT job in which the source segments in $\mathrm{TM_{trans}}$ are translated using the TUs in $\mathrm{TM_{test}}$. For each source segment in $\mathrm{TM_{trans}}$, and using the same FMS threshold Θ used during training, we computed the set of the matching TUs in $\mathrm{TM_{test}}$, and classified the words in their target segments as "keep" or "change".

We used the general-purpose free/open-source MT system `apertium-en-es`, version 0.7, built upon version 3.2 of the Apertium free/open-source MT platform[2] (Forcada et al., 2011). This is a rule-based MT system that achieves a BLEU score (Papineni et al., 2002) of 0.20 on the test set provided for the WMT10 translation task.[3] This BLEU score may be considered low compared to those achieved by other MT systems, ranging around 0.27, for the same language pair and on the same test set (Callison-Burch et al., 2010).

4.1 Corpora

The TMs we have used were extracted from two different parallel corpora already aligned at the sentence level: the JRC-Acquis corpus version 3 (Steinberger et al., 2006),[4] which contains the total body of European Union law, and the EMEA corpus version 0.3 (Tiedemann, 2009),[5] which is a compilation of documents from the European Medicines Agency. Before extracting the TMs used, both corpora were tokenized and lowercased, and then sentence pairs in which either of the sentences was empty, was more than 9 times longer than its counterpart, or was longer than 40 words were removed.

The sentences in $\mathrm{TM_{trans}}$, $\mathrm{TM_{test}}$ and $\mathrm{TM_{in}}$ were randomly chosen without repetition from the JRC-Acquis corpus. $\mathrm{TM_{test}}$ and $\mathrm{TM_{in}}$ consist of 10,000 parallel sentences each, whereas $\mathrm{TM_{trans}}$ consists of 5,000 sentence pairs. $\mathrm{TM_{out}}$ consists of 10,000 sentence pairs randomly chosen without repetition from the EMEA corpus. Table 1 summarizes this information. It is important to remark that these TMs may contain incorrect TUs as a result of wrong sentence alignments, and that true TMs (as corrected by professional translators) are not usually so noisy.

TM	TUs	Corpus
$\mathrm{TM_{out}}$	10,000	EMEA
$\mathrm{TM_{in}}$	10,000	JRC-Acquis
$\mathrm{TM_{test}}$	10,000	JRC-Acquis
$\mathrm{TM_{trans}}$	5,000	

Table 1: Data about the TMs used in the evaluation.

$\Theta(\%)$	TUs	N_{words}
50	9.5	484,523
60	6.0	303,193
70	4.5	220,304
80	3.5	166,762
90	0.9	42,708

Table 2: Average number of matching TUs per segment and number of words to classify for different FMS thresholds (Θ).

As regards the number of TUs found in $\mathrm{TM_{test}}$ when simulating the translation of the SL segments in $\mathrm{TM_{trans}}$, Table 2 reports, for the different FMS thresholds we have used, the average number of TUs per segment to be translated and the total number of words to be classified. These data provide an idea of the repetitiveness of the corpora used to carry out the experiments.

4.2 Baseline

We compare the performance of our MT-based approach to two different baselines. The first one corresponds to the approach by Esplà et al. (2011) described in the introduction. To reproduce the experiments in (Esplà et al., 2011), we trained three different SWA models, each one on a different TM: $\mathrm{TM_{in}}$, $\mathrm{TM_{out}}$ and $\mathrm{TM_{test}}$. In addition, we used the union of the SL–TL and TL–SL alignments to maximize the coverage, i.e. the proportion of words for which a recommendation is made. The second baseline is a *naïve* one which bases its recommendations only on the FMS. This baseline uses the perceptron classifier defined in Section 3 but without the features coming from MT, so it only uses one feature:

[2] `http://www.apertium.org`
[3] `http://www.statmt.org/wmt10/`
[4] `http://wt.jrc.it/lt/Acquis/`
[5] `http://opus.lingfil.uu.se/EMEA.php`

the FMS between s' and s_i. In this way, the improvement provided by the MT-based features can be directly measured. The classifier was trained on both $\mathrm{TM}_{\mathrm{in}}$ and $\mathrm{TM}_{\mathrm{out}}$ to obtain the corresponding set of weights.

4.3 Evaluation

To test our approach we computed, for each source segment s'_m in $\mathrm{TM}_{\mathrm{trans}}$, the set of matching TUs $\{(s_i, t_i)\}_{i=1}^{N}$ in $\mathrm{TM}_{\mathrm{test}}$ whose FMS is above threshold Θ. We then used the classifier trained using the same threshold Θ to calculate the keeping probability $p_K(w_{ij}, s'_m, s_i, t_i)$ for each word w_{ij} in t_i through eq. (1).

The *accuracy* attained by a classifier for the test set is computed as $N_{\mathrm{correct}}/N_{\mathrm{classified}}$, where $N_{\mathrm{classified}}$ is the total number of words classified, and

$$N_{\mathrm{correct}} = \sum_{m=1}^{N_T} \sum_{i \in \mathrm{match}(m,\Theta)} \sum_{j=1}^{|t_i|} S_{ijm} \qquad (5)$$

where $S_{ijm} = 1$ if

$$p_K(w_{ij}, s'_m, s_i, t_i) - \hat{p_K}(w_{ij}, s'_m, s_i, t_i) < \frac{1}{2}$$

and zero otherwise; $\hat{p}_K$ is calculated through eq. (2) as during training. Note that, while the MT-based approach and the naïve baseline are always able to provide a recommendation, the SWA-based approach does not provide a recommendation for the unaligned words. The *coverage*, that is, the fraction of words classified by the SWA-based baseline is defined as $N_{\mathrm{classified}}/N_{\mathrm{words}}$.

It is worth noting that eq. (5) above may also be seen as an error function such as eq. (3) and could have been directly used to train the classifier, but not using a gradient descent algorithm, as it is not differentiable. We tested the optimization of eq. (5) by using a multidimensional simplex algorithm (Nelder and Mead, 1965) and found out that the experimental results obtained were overall slightly worse.

5 Results and discussion

We trained and evaluated the parametric classifier introduced in Section 3 for the MT-based approach when considering different FMS thresholds and maximum sub-segment lengths L in the interval $[1, 5]$. In this section we only report the results achieved with $L = 3$, that is, for sub-segment lengths $n \in [1, 3]$, because this is the maximum sub-segment length for which the best results were obtained.

Table 3 reports, for different FMS thresholds Θ, the accuracy and coverage achieved by the SWA-based baseline when the alignment probabilities used to align the words in the parallel sentences in $\mathrm{TM}_{\mathrm{test}}$ are obtained from $\mathrm{TM}_{\mathrm{in}}$, $\mathrm{TM}_{\mathrm{out}}$ and $\mathrm{TM}_{\mathrm{test}}$ itself, and the accuracy achieved by the naïve baseline. The results obtained by the naïve baseline when it is trained on $\mathrm{TM}_{\mathrm{in}}$ and on $\mathrm{TM}_{\mathrm{out}}$ are reported in the same column because they are exactly the same. This is because this baseline is so basic that it always classifies the words as "keep". Table 4 reports the accuracy achieved by our MT-based approach when the classifier is trained on both $\mathrm{TM}_{\mathrm{in}}$ and $\mathrm{TM}_{\mathrm{out}}$, respectively. Note that the coverage of both the naïve baseline and our MT-based approach equals 100% in all cases. Both accuracy and coverage are reported together with their confidence intervals for a statistical significance level $p = 0.99$ (DeGroot and Schervish, 2002, Sec. 7.5).

As expected, both the SWA-based baseline and the MT-based approach outperform the naïve baseline. As can be seen, for the in-domain $\mathrm{TM}_{\mathrm{in}}$ the accuracy of the MT-based perceptron classifier is very similar to that obtained by the SWA-based baseline, being worse only when Θ is under 60%. In this regard, it is important to note that professional translators tend to use values for Θ above 60% (Bowker, 2002, p. 100). The results achieved when $\mathrm{TM}_{\mathrm{out}}$ is used for training show that our MT-based approach outperforms the SWA-based baseline for FMS thresholds above 60%. In addition, the coverage of the SWA-based baseline drops off when the alignment probabilities are obtained from $\mathrm{TM}_{\mathrm{out}}$.

The best results for the SWA-based baseline are obtained, as expected, when trained on $\mathrm{TM}_{\mathrm{test}}$ itself: the accuracy is very similar to that obtained when training on $\mathrm{TM}_{\mathrm{in}}$, and coverage approaches 100%. Note that the results of the baseline trained on $\mathrm{TM}_{\mathrm{test}}$ and of our MT-based approach cannot be directly compared because they were not trained on the same TM. Moreover, we think that the use of $\mathrm{TM}_{\mathrm{test}}$ for training does not represent a real use scenario, since TMs are not always static and new TUs are usually added to them during a translation job.

For FMS thresholds higher than 60%, the accuracy of the perceptron classifier is almost the same independently of the TM ($\mathrm{TM}_{\mathrm{in}}$ or $\mathrm{TM}_{\mathrm{out}}$) used for

Θ (%)	SWA-based baseline						Naïve baseline
	$\mathrm{TM_{in}}$		$\mathrm{TM_{out}}$		$\mathrm{TM_{test}}$		$\mathrm{TM_{in}}$ / $\mathrm{TM_{out}}$
	Acc. (%)	Cover. (%)	Acc. (%)	Cover. (%)	Acc. (%)	Cover. (%)	Acc. (%)
50	90.37 ± .11	95.62 ± .08	87.01 ± .13	91.57 ± .10	90.50 ± .11	99.26 ± .03	67.62 ± .17
60	93.36 ± .12	95.02 ± .10	90.60 ± .14	91.48 ± .13	93.30 ± .12	99.43 ± .04	75.42 ± .20
70	96.60 ± .10	94.41 ± .13	94.23 ± .13	91.15 ± .16	96.27 ± .10	99.61 ± .03	78.37 ± .23
80	98.02 ± .09	93.80 ± .15	95.55 ± .14	90.63 ± .18	97.58 ± .10	99.76 ± .03	80.60 ± .25
90	97.73 ± .19	93.73 ± .30	97.70 ± .20	90.83 ± .36	97.46 ± .20	99.74 ± .06	87.77 ± .41

Table 3: For different FMS thresholds (Θ): accuracy (Acc.) and coverage (Cover.) of word-keeping recommendation obtained by the SWA-based baseline when the alignment probabilities used to align the words in the parallel sentences in $\mathrm{TM_{test}}$ are obtained from $\mathrm{TM_{in}}$, $\mathrm{TM_{out}}$ and $\mathrm{TM_{test}}$ itself, and accuracy obtained by the naïve baseline.

Θ (%)	$\mathrm{TM_{in}}$	$\mathrm{TM_{out}}$
50	88.97 ± .12	85.29 ± .13
60	93.46 ± .12	93.46 ± .12
70	96.46 ± .10	96.46 ± .10
80	98.16 ± .09	98.16 ± .09
90	97.80 ± .18	97.80 ± .18

Table 4: For different FMS thresholds (Θ): accuracy obtained by our MT-based approach for the perceptron classifier when it is trained on the in-domain $\mathrm{TM_{in}}$ and on the out-of-domain $\mathrm{TM_{out}}$.

training. This results show that once the classifier has been trained, one can use its parameters for word-keeping recommendation to translate a text from a different domain.

An in-depth analysis of the feature weights obtained with both TMs used for training shows some regularities. In both cases the weight assigned to the FMS feature decreases as the value of Θ increases, which seems reasonable given the fact that the different values that the FMS can take is reduced as Θ grows. Conversely, the value of λ_0, i.e. the bias of the perceptron, gets higher as the value of Θ increases, an indication that in those cases it is more likely that the words are classified as "keep"; i.e., the higher the FMS, the less words to change.

With respect to the rest of feature weights, they take positive values for positive evidences and negative values for negative evidences, which means that these features are opposing (as expected). In general, the absolute value for the weights decreases with the length of the segments from which they have been obtained. This means that the shortest sub-segments provide more information than the longest ones. This may be explained by the fact that long sub-segments are more likely to contain unmatched words, and therefore, when features are computed for a word in one of these long sub-segments, it is more likely to get negative evidence even if the word should be kept.

6 Concluding remarks

In this paper we have presented a novel approach to assist CAT users using TMs by recommending them which target-side words in the TUs proposed by the CAT system have to be changed or kept unedited. The method we propose imposes no constraints on the type of MT system to use, and may use more than an MT system at the same time, or even other bilingual resources, to obtain a set of features that are then combined through a binary classifier to determine the words to be changed or kept unedited. In any case, MT is never used to actually translate any new material.

Our results show that the parameters of the binary classifier are basically domain-independent. This implies that it is neither necessary to re-train the classifier for each new TM, nor to take into account the newly created TUs. CAT users can therefore use our MT-based approach provided that they have the classifier, a suitable set of feature weights and access to the MT system used for training. We plan to study the dependency of parameters on the MT system used and on the language pair.

The experiments conducted attest to the feasibility of the method, and open up pathways for future work such as extending the method so as to be able not only to recommend the user which words to keep unedited, but also to suggest a translation for the words to change; trying alternative parametric classifiers; and performing experiments to measure the improvement in the productivity of human translators using a TM-based CAT system integrating our MT-based word-keeping recommendation system (for in-

stance by integrating this method in the free/open-source CAT system OmegaT,[6] which already has support to interface with on-line MT).

Acknowledgements: Work supported by Spanish government through TIN2009-14009-C02-01 project. M.L. Forcada's sabbatical stay at Dublin City University was supported by Science Foundation Ireland (SFI) through ETS Walton Award 07/W.1/I1802 and by Universitat d'Alacant (Spain). The authors thank Juan Antonio Pérez-Ortiz, Andy Way and Harold Somers for suggestions.

References

E. Biçici and M. Dymetman. 2008. Dynamic translation memory: Using statistical machine translation to improve translation memory fuzzy matches. In *Computational Linguistics and Intelligent Text Processing*, volume 4919 of *LNCS*, pages 454–465. Springer.

L. Bowker, 2002. *Computer-aided translation technology: a practical introduction*, chapter Translation-Memory Systems, pages 92–127. University of Ottawa Press.

C. Callison-Burch, P. Koehn, C. Monz, K. Peterson, M. Przybocki, and O. Zaidan. 2010. Findings of the 2010 Joint Workshop on Statistical Machine Translation and Metrics for Machine Translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, Uppsala, Sweden.

M. H. DeGroot and M. J. Schervish. 2002. *Probability and Statistics*. Addison-Wesley, third edition.

R.O. Duda, P.E. Hart, and D.G. Stork. 2000. *Pattern Classification*. John Wiley and Sons Inc., second edition.

M. Esplà, F. Sánchez-Martínez, and M.L. Forcada. 2011. Using word alignments to assist computer-aided translation users by marking which target-side words to change or keep unedited. In *Proceedings of the 15th Conference of the European Association for Machine Translation*, pages 81–88, Leuven, Belgium.

M.L. Forcada, M. Ginestí-Rosell, J. Nordfalk, J. O'Regan, S. Ortiz-Rojas, J.A. Pérez-Ortiz, F. Sánchez-Martínez, G. Ramírez-Sánchez, and F.M. Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation. Special Issue on Free/Open-Source Machine Translation*, in press.

P. Koehn and J. Senellart. 2010. Convergence of translation memory and statistical machine translation. In *Proceedings of the Second Joint EM+/CNGL, Workshop "Bringing MT to the User: Research on Integrating MT in the Translation Industry"*, pages 21–31, Denver, USA.

[6] `http://www.omegat.org`

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

L. Kranias and A. Samiotou. 2004. Automatic translation memory fuzzy match post-editing: A step beyond traditional TM/MT integration. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon, Portugal.

E. Lagoudaki. 2008. The value of machine translation for the professional translator. In *Proceedings of the 8th Conference of the Association for Machine Translation in the Americas*, pages 262–269, Waikiki, USA.

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

A. Meyers, M. Kosaka, and R. Grishman. 1998. A multilingual procedure for dictionary-based sentence alignment. In *Machine Translation and the Information Soup*, volume 1529 of *LNCS*, pages 187–198. Springer.

J.A. Nelder and R. Mead. 1965. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313.

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

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, USA.

R. Sikes. 2007. Fuzzy matching in theory and practice. *MultiLingual*, 18(6):39–43.

M. Simard and P. Isabelle. 2009. Phrase-based machine translation in a computer-assisted translation environment. In *Proceedings of the 12th Machine Translation Summit*, pages 120–127, Ottawa, Canada.

H. Somers, 2003. *Computers and translation: a translator's guide*, chapter Translation Memory Systems, pages 31–48. John Benjamins Publishing Company, Amsterdam, Netherlands.

R. Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, and D. Tufiş. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 2142–2147, Genoa, Italy.

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

V. Zhechev and J. van Genabith. 2010. Seeding statistical machine translation with translation memory output through tree-based structural alignment. In *Proceedings of the COLING'10, Workshop on Syntax and Structure in Statistical Translation*, pages 43–51, Beijing, China.