Instance Selection for Online Automatic Post-Editing in a Multi-domain Scenario

Rajen Chatterjee chatterjee@fbk.eu
University of Trento, Trento, Italy
Fondazione Bruno Kessler, Trento, Italy

Mihael Arcan mihael.arcan@insight-centre.org
Insight Centre for Data Analytics
National University of Ireland, Galway, Ireland

Matteo Negri negri@fbk.eu
Fondazione Bruno Kessler, Trento, Italy

Marco Turchi turchi@fbk.eu
Fondazione Bruno Kessler, Trento, Italy

Abstract

In recent years, several end-to-end online translation systems have been proposed to successfully incorporate human post-editing feedback in the translation workflow. The performance of these systems in a multi-domain translation environment (involving different text genres, post-editing styles, machine translation systems) within the automatic post-editing (APE) task has not been thoroughly investigated yet. In this work, we show that when used in the APE framework the existing online systems are not robust towards domain changes in the incoming data stream. In particular, these systems lack in the capability to learn and use domain-specific post-editing rules from a pool of *multi-domain* data sets. To cope with this problem, we propose an online learning framework that generates more reliable translations with significantly better quality as compared with the existing online and batch systems. Our framework includes: *i)* an instance selection technique based on information retrieval that helps to build domain-specific APE systems, and *ii)* an optimization procedure to tune the feature weights of the log-linear model that allows the decoder to improve the post-editing quality.

1 Introduction

Nowadays, machine translation (MT) is a core element in the computer-assisted translation (CAT) framework. The motivation for integrating MT in the CAT framework lies in its capability to provide useful suggestions for unseen segments, which helps to increase the translators' productivity. However, it has been observed that MT is often prone to systematic errors that human post-editing has to correct before publication. The by-product of this "translation as post-editing" process is an increasing amount of parallel data consisting of MT output on one side and its corrected version on the other side. This data can be leveraged to develop automatic post-editing (APE) systems capable not only to spot recurring MT errors, but also to correct them. Thus, integrating an APE system inside the CAT framework can further improve the quality of the suggested segments, reduce the workload of human post-editors and increase the productivity of the translation industry. As pointed out in (Parton et al., 2012) and (Chatterjee et al., 2015b), from the application point of view APE components would make it possible to:

- Improve the MT output by exploiting information unavailable to the decoder, or by performing deeper text analysis that is too expensive at decoding stage;
- Cope with systematic errors of an MT system whose decoding process is not accessible;
- Provide professional translators with improved MT output quality to reduce (human) PE effort;
- Adapt the output of a general-purpose MT system to the lexicon/style requested in a specific application domain.

In the last decade several works have shown that the quality of the machine translated text can be improved significantly by post-processing the translations with an APE system (Simard et al., 2007a; Dugast et al., 2007; Terumasa, 2007; Pilevar, 2011; Béchara et al., 2011; Chatterjee et al., 2015b, 2016). These systems mainly follow the phrase-based machine translation approach where the MT outputs (with optionally the source sentence) are used as the source language corpus and the post-edits are used as the target language corpus. A common trait of all these APE systems is that they were developed in a batch mode, which consists of training the models over a batch of parallel sentences, optimizing the parameters over a development set, and then decoding the test data with the tuned parameters. Although these standard approaches showed promising results, they lack the ability to incorporate human feedback in a real-time translation workflow. This led to the development of online learning algorithms that can leverage the continuous streams of data arriving in the form of human post-editing feedback to dynamically update the models and tune the parameters on-the-fly within the CAT framework. In recent years, several online systems have been proposed in MT (see Section 2 for more details) to address the problem of incremental training of the models or on-the-fly optimization of feature weights. Few online MT systems have also been applied to the APE scenario (Simard and Foster, 2013; Lagarda et al., 2015) in a controlled working environment in which the systems are trained and evaluated on homogeneous/coherent data where the training and test sets share similar characteristics. Moving from this controlled lab environment to real-world translation workflow, where training and test data can be produced by different MT systems, post-edited by various translators and belong to several text genres, makes the task more challenging, because the APE systems have to adapt to all these diversities in real-time. We define this scenario as a *multi-domain* translation environment (MDTE), where a domain is made of segments belonging to the same text genre and the MT outputs are generated by the same MT system. To reproduce this scenario, in our experiments we run the online APE systems on the concatenation of two datasets belonging to different domains.

A preliminary evaluation in the MDTE scenario reveals that online systems are not robust enough to learn and adapt towards the dynamics of the data, mainly because they try to leverage all the seen data without considering the peculiarities of each domain. In the long-run, these systems tend to become more and more generic, which may not be useful and even harmful to automatically post-edit domain-specific segments. To address this problem, for the first time, we propose an online APE system that is able to efficiently work in a MDTE scenario. Our intuition is that an online APE model trained with few but relevant data (with respect to the segment to be post-edited) can be more reliable than using all the available data *as-is*. To validate this intuition, we propose an online APE system based on an instance selection (IS) technique that is able to retrieve the most relevant training instances from a pool of multi-domain data for each segment to post-edit. The selected data are then used to train and tune the APE system on-the-fly. The relevance of a training sample is measured by a similarity score that takes into account the context of the segment to be post-edited. This technique allows our online APE system to be flexible enough to decide if it has the correct knowledge for post-editing a sentence or if it is safer to keep the MT output untouched, avoiding possible damages of correction made with insufficient/unreliable knowledge. The results of our experiments with various data sets show that our online learning approach based on IS is: *i)* able to outperform the batch and the other online APE techniques in the single domain scenario, and *ii)* robust enough to work in a MDTE to generate reliable post-edits with significantly better performance than the existing online APE systems.

2 Online Translation Systems

Online translation systems aim to incorporate human post-editing feedback (or the corrected version of the MT output) into their models in real-time, as soon as it becomes available. This feedback helps the system to learn from the mistakes made in the past translations and to avoid repeating them in future translations. This continuous learning capability will eventually improve the quality of the translations and consequently increase the productivity of the translators/post-editors (Tatsumi, 2009) working with MT suggestions in a CAT environment. The basic workflow of an online translation system goes through the following steps repeatedly: *i)* the system receives an input segment; *ii)* the input segment is translated and provided to the post-editor to fix any errors in it; and *iii)* the human post-edited version of the translation is incorporated back into the system, by stepwise updating the underlying models and parameters. In the APE context, the input is a machine-translated segment (optionally with its corresponding source segment), which is processed by the online APE system to fix errors, and then verified by the post-editors. Several online translation systems have been proposed over the years (Hardt and Elming, 2010; Bertoldi et al., 2013; Mathur et al., 2013; Simard and Foster, 2013; Ortiz-Martınez and Casacuberta, 2014; Denkowski et al., 2014; Wuebker et al., 2015, inter alia). In this section, we describe two online systems that have been used in the APE task (PEPr, and Thot), and one in the MT scenario which is similar to our proposed system (Realtime cdec):

PEPr: Post-Edit Propagation: Simard and Foster (2013) proposed a method for post-edit propagation (PEPr), which learns post-editors' corrections and applies them on-the-fly to further MT output. Their proposal is based on a phrase-based SMT system, used in an APE setting with online learning mechanism. To perform post-edit propagation, this system was trained incrementally using pairs of machine-translated *(mt)* and human post-edited *(pe)* segments as they were produced. When receiving a new pair *(mt, pe)*, word alignments are obtained by using Damerau-Levenshtein distance. In the next step the phrase pairs are extracted and appended to the existing phrase table. The whole process is assumed to take place within the context of a single document. For every new document the APE system begins with an "empty" model. Since the post-editing rules are learned for a given document they can be more precise and useful for that document, but the limitation is that knowledge gained after processing one document is not utilized for other similar documents. This limitation can be addressed by our system (Section 3), in which we maintain one global knowledge base to store all the processed documents, still being able to retrieve post-editing rules specific to a document to be translated.

Thot: The Thot toolkit (Ortiz-Martınez and Casacuberta, 2014) is developed to support fully automatic and interactive statistical machine translation.[1] It was also used by Lagarda et al. (2015) in an online setting for the APE task, to perform large-scale experiments with several

[1] https://github.com/daormar/thot

data sets for multiple language pairs, with base MT systems built using different technologies (rule-based MT, statistical MT). In the majority of their experiments online APE successfully improved the quality of the translations obtained from the base MT system by a significant margin. To update the underlying translation and language models with the user feedback, a set of sufficient statistics was maintained that can be incrementally updated. In the case of language model, only the n-gram counts are required to maintain sufficient statistics. To update the translation model, an incremental version of EM algorithm is used to first obtain word alignment and then phrase pairs counts were extracted to update the sufficient statistics. Other features like source/target phrase-length models or distortion model are implemented by means of geometric distributions with fixed parameters. The sentence length model is implemented by means of Gaussian distributions. However, the feature weights of the log-linear model are static throughout the online learning process, as opposed to our method that updates the weights on-the-fly. Also, this method learns post-editing rules from all the data processed in real-time, whereas, our approach learns from the most relevant data points.

Realtime cdec: Denkowski et al. (2014) proposed an online model adaptation method to leverage human post-edited feedback to improve the quality of an MT system in a real-time translation workflow. To build the translation models they use a static suffix array (Zhang and Vogel, 2005) to index initial data (or a seed corpus), and a dynamic lookup table to store information from the post-edited feedback. To decode a sentence, the statistics of the translation options are computed both from the suffix array and from the lookup table. An incremental language model is maintained and updated with each incoming human post-edit. To update the feature weights they used an extended version of the margin-infused relaxed algorithm (MIRA) (Chiang, 2012). The decoding is treated as simply the next iteration of MIRA, where a segment is first translated and then its corresponding reference/post-edition is provided to the model, and MIRA updates the parameters. While this system was earlier used in the context of MT, in this work we use it to investigate its applicability in online APE. A key difference between this approach and ours is the sampling technique. The former uses suffix-arrays to always retrieve the top *k* source phrases, whereas in our approach the number of samples (or the training instances) is dynamically set to use only the most relevant ones. Another difference is visible in the parameter optimization step. Realtime cdec optimizes the feature weights of the log-linear model after decoding each segment, whereas, our method optimizes the weights specifically for the segment to be post-edited.

3 Instance Selection for Online APE System

The online systems described in Section 2 compute and update the feature scores of the log-linear models based on all the previously seen data. This indicates that, in the long-run, the model will tend to become more and more generic, since the data processed in the online scenario may belong to multiple domains as explained in Section 1. Having a generic model might not be useful to retrieve the domain-specific post-editing rules needed to fix errors in a particular document. One solution is to build document-specific APE models as proposed by Simard and Foster (2013). In their approach, however, once the entire document is processed the models are reset back to their original state, due to which the knowledge gained from the current document is lost. To preserve all the knowledge gained in the online learning process, at the same time being able to apply specific post-editing rules when needed, we propose an instance selection technique for online APE. Our proposed framework, as shown in Figure 1, uses a global knowledge base to preserve all the data points seen in the online process, and has the ability to retrieve specific data points whose context is similar to the segment to be post-edited. These data points are used to build reliable APE models. When there are no reliable data points in the knowledge base, the MT output is kept untouched, as opposed to the existing APE systems, which tend to

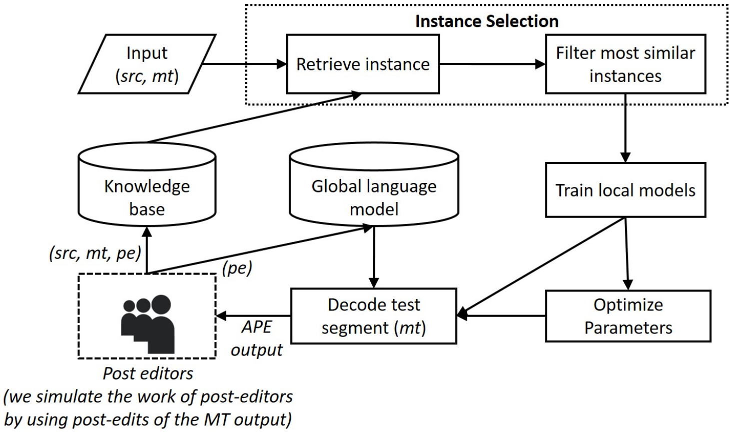

Figure 1: Architecture of our online APE system

always "translate" the given input segment independently from the reliability of the applicable correction rules. This approach of post-editing with reliable information only makes our system more precise compared with others (see results in Section 5): that is when a post-editing rule is applied it is more likely to improve the quality of the translation. When no reliable knowledge is available for the correction, the MT output is left untouched.

We propose online APE but we actually "emulate" it by processing the data points one at a time. Our proposed algorithm assumes to have the following data to run the online experiments: *i)* source (*src*); *ii)* MT output (*mt*); and *iii)* human post-edits (*pe*) of the MT output. At the beginning the knowledge base of our online APE system is empty and it will be updated whenever a new instance (a tuple containing parallel segments from all the above mentioned documents) is processed. When the system receives an input (*src*, *mt*), it proceeds through the following steps:

Instance Selection. Initially, it selects the most relevant training instances from a pool of *multi-domain* data stored in our knowledge base. This will help to build a reliable APE model for each input segment processed in real-time. The relevance of the training instances with respect to the input segment is measured in terms of a similarity score based on the term frequency—inverse document frequency (*tf-idf*), generally used in information retrieval. The larger the number of words in common between the training and the input sentences, the higher is the score. In our system, these scores are computed using the Lucene library.[2] Only those training instances that have a similarity score above a certain threshold (decided over a held-out development set) are used to build the system. In case there are no training instances available, we preserve the input segment as it is. Indeed, we assume that APE with unreliable information can damage the *mt* segment instead of improving the translation quality. This is one of the main outcomes of the first APE pilot task organized last year within the WMT initiative (Bojar et al., 2015) and, as we will see from our results, it represents a major problem for the approaches that always translate the given input segments. The proposed instance selection technique (or sampling mechanism) differs from the one proposed in real-time cdec (Denkowski et al., 2014), which uses suffix-arrays to select the top *k* instances. In our approach the sample size is in fact dynamically set in order to select only the most similar ones. This allows us to build more reliable models (since the underlying data better resembles the test segment), and to gain speed

[2] https://lucene.apache.org/

when the sample size is small. The use of a *tf-idf* similarity measure was proposed before in the context of machine translation by Hildebrand et al. (2005) to create a pseudo in-domain corpus from a big out-of-domain corpus. Our work is the first to investigate it for the APE task in an online learning scenario.

Model Creation. From the selected instances we build several local models. The first is the language model: A tri-gram local language model is built over the target side of the training corpus with the IRSTLM toolkit (Federico et al., 2008). Since the selected training data closely resembles the input segment, we believe that the local LM can capture the peculiarities of the domain to which the input segment belongs. Along with the local LM we always use a tri-gram global LM, which is updated whenever a human post-edition (*pe*) is received. The other local models are the translation and the reordering models: these local models are built over the training instances retrieved from the knowledge base. Since the training instances are very similar to the input segment, the post-editing rules learned from these local models are more reliable for the test segment. These models are build with the Moses toolkit (Koehn et al., 2007) and the word alignment of each sentence pair is computed using the incremental GIZA++ software.[3]

Parameter Optimization. The parameters are optimized over a section of the selected instances (development set). The size of this development set is critical: if it is too large, then the parameter optimization will be expensive. On the other hand, if it is too small the tuned weights might not be reliable. To achieve fast optimization with reliably-tuned weights, multiple instances of MIRA are run in parallel on several small development sets and all the resulting weights are then averaged. For this purpose, the data selected by the instance selection module are randomly split in training and development sets three times. A minimum number of selected sentence pairs is required to trigger the parameter optimisation process. If this minimum value is not reached, the optimization step is skipped because having few sentences might not yield to reliable weights. In this case, the weights computed on the previous input segment are used. In our experiments, we observed that this solution is more reliable and efficient than the feature weights obtained with a single tuning, as it was previously proposed in (Cettolo et al., 2011). We believe this procedure to optimize the feature weights over a development set that closely resembles the test segment can help to obtain weights more suitable to the segment to be post-edited.

Decode Test Segment. To decode the input segments, all the local models (language, translation, reordering) are built with all the selected instances. The log-linear feature weights are computed by taking the arithmetic mean of the tuned weights for the three data splits. The decoding process is performed with the Moses toolkit recalling that the input segment is kept untouched when no reliable information is available in the knowledge base.

Update Global Repository. In a real translation workflow, the automatically post-edited version (or the MT output, if there were no training data available) is provided to a post-editor for correction, and the corrected version is incorporated back into the system. To avoid the unnecessary costs of involving human post-editors in-the-loop when running these experiments, we simulate this condition by using the human post-edits of the MT output (which are already available in the data set). Each newly processed instance is added to our knowledge base, and the global language model is updated with the post-edited segment.

[3] `https://code.google.com/archive/p/inc-giza-pp/`

4 Experimental Setup

4.1 Data

To examine the performance of the online APE systems in a multi-domain translation environment, we select two data sets for the English-German language pair belonging to the information technology (IT) domain. Although they come from the same domain (IT), they feature variability in terms of vocabulary coverage, MT errors, and post-editing style. The two data sets are respectively a subset of the Autodesk Post-Editing Data corpus [4] and the resources used at the second round of the APE shared task at the First Conference on Machine Translation (WMT2016) (Bojar et al., 2016).[5] The data sets are pre-processed to obtain a joint-representation that links each source word with a MT word (*mt#src*). This representation has been proposed in the context-aware APE approach by Béchara et al. (2011) and leverages the source information to disambiguate post-editing rules. Recently, Chatterjee et al. (2015b) also confirmed this approach to work better than translating from raw MT segments over multiple language pairs. The joint-representation is used as a source corpus to train all the APE systems reported in this paper and it is obtained by first aligning the words of source (*src*) and MT (*mt*) segments using MGIZA++ (Gao and Vogel, 2008), and then each *mt* word is concatenated with its corresponding *src* words.

The Autodesk training, development, and test sets consist of 12,238, 1,948, and 1,956 segments respectively, while the WMT2016 data contains 12,000, 1,000, and 2,000 segments. Table 1 provides some additional statistics of the source (*mt#src*) and target (*pe*) training corpus, the repetition rate (RR) to measure the repetitiveness inside a text (Bertoldi et al., 2013), and the average TER score for both the data sets (computed between MT and PE). It is interesting to note that the Autodesk data set has on average shorter segments compared with the WMT2016 corpus. This suggests that learning and applying post-editing rules in the Autodesk corpus can be easier than using the WMT2016 segments, because dealing with long segments generally increases the complexity of the rules extraction and decoding processes. Moreover, the WMT2016 data set has a repetition rate similar to the Autodesk even though it has more tokens. This indicates that the data is more sparse raising the difficulty of extracting reliable post-editing rules. Looking at the TER score, the smaller value of the WMT2016 data set compared with the Autodesk one suggests that the room for improvement is lower, because there are less corrections to perform and the chance to deteriorate the original MT output is larger.

	Tokens		Types		Avg. segment length		RR	TER
	mt#src	*pe*	*mt#src*	*pe*	*mt#src*	*pe*	(*mt#src*)	
Autodesk	153,943	160,801	31,939	15,023	12.57	13.13	4.938	45.35
WMT2016	210,573	214,720	32211	16,388	17.54	17.89	4.907	26.22

Table 1: Data statistics

The diversity of the two data sets is further measured by computing the vocabulary overlap between the two joint-representations. This is performed internally to each data set (splitting the training data in two halves) and across them. As expected, in the first case the vocabulary overlap is much larger ($> 40\%$) than in the second one ($\sim 15\%$), and this indicates that the two data sets are quite different and few information can be shared. All the aforementioned aspects show the large variability in the corpora making them suitable to emulate the multi-domain translation environment.

[4] https://autodesk.app.box.com/v/autodesk-postediting

[5] http://www.statmt.org/wmt16/ape-task.html

4.2 Evaluation metrics

The performance of the different APE systems is evaluated using three different metrics: Translation Error rate (TER) (Snover et al., 2006), BLEU (Papineni et al., 2002) and Precision (Chatterjee et al., 2015a). TER and BLEU measure the similarity between the MT outputs and their references by looking at n-grams overlap (TER at word level, BLEU from 1 to 4 words). To give a better insight on the APE performance, we also report Precision, computed as the ratio of the number of sentences an APE system improves (with respect to the MT output) over all the sentences it modifies.[6] Values larger than 50% indicate that the APE system is able to improve the quality of most of the sentences it changes.

Statistical significance tests are computed using the paired bootstrap resampling technique (Koehn, 2004) for the BLEU metric and the stratified approximate randomization test (Clark et al., 2011) for TER.

4.3 Terms of comparison

We evaluate our online learning approach against four different terms of comparison.

MT. Our baseline is the *"do-nothing"* system that simply returns the MT outputs without changing them. As discussed in (Bojar et al., 2015), this baseline can be particularly hard to beat when the repetition rate of the data is low and due to the tendency of the APE systems to over-correct the MT output.

Batch APE. This APE system is developed in a batch mode following the approach proposed in Chatterjee et al. (2015b). It is similar to the context-aware method (Béchara et al., 2011), but it uses word alignments produced by the monolingual machine translation APE technique proposed in Simard et al. (2007b). Being a batch method, it cannot learn from the test set, but it leverages all the training points at the same time.

Online APE. We compare our approach against two online systems: *i)* the Thot toolkit that had been previously used in the online APE task, and *ii)* Realtime cdec that, among the other online MT systems, is the closest to our approach (*i.e.* it uses a data selection mechanism), but has never been tested in the APE scenario. Another online APE approach is PEPr that was meant for document level APE, but since we are working with data sets that do not have any intrinsic document structure, we do not find it to be a suitable term of comparison.

5 Experiments and Results

Our preliminary objective is to examine if the online learning methods are able to achieve results that are competitive with those of batch methods, which are potentially favored by the possibility to leverage all the training data at the same time. For this test, all the algorithms are evaluated in the classic *in-domain* setting, where training, development, and test sets are sampled from the same data set or domain. All the online APE methods are run in two modes; *i)* batch: the test set is not used in the learning process (to have a fair comparison with the batch APE), *ii)* online: the test set is leveraged in the online learning process. The experiments are performed for both the data sets (Autodesk and WMT2016), and their corresponding results are reported in Table 2 and Table 3 respectively. The parameters of our approach (*i.e.* similarity score threshold and minimum number of selected sentence) are optimised on the development set following a grid search strategy. We set the threshold values to 0.8 and 1 respectively for the Autodesk and WMT2016 datasets and the minimum number of selected sentences to 20.

[6] For each sentence in the test set, if the TER score of the APE output is different than the TER score of the MT then the sentence is considered as a modified sentence

	Batch mode			Online mode		
	BLEU	TER	Precision (%)	BLEU	TER	Precision (%)
MT	39.28	46.48	N/A	N/A	N/A	N/A
Batch APE	44.14	43.24	61.34	N/A	N/A	N/A
cdec	43.13†	43.86†	54.22	43.19†	43.69†	54.75
Thot	43.21†	44.70†	55.69	43.34†	44.62†	56.27
Our approach	**44.68**†	**41.98**†	**79.26**	**44.76**†	**41.95**†	**79.20**

Table 2: Autodesk in-domain (†: statistically significant wrt. Batch APE with $p<0.05$)

	Batch mode			Online mode		
	BLEU	TER	Precision (%)	BLEU	TER	Precision (%)
MT	62.11	24.76	N/A	N/A	N/A	N/A
Batch APE	63.06	25.07	48.55	N/A	N/A	N/A
cdec	61.99†	25.26	45.17	61.80†	25.35†	42.83
Thot	62.06†	25.26	42.92	62.22†	25.22	43.69
Our approach	**62.97**	**24.53**†	**61.46**	**63.19**	**24.39**†	**62.62**

Table 3: WMT2016 in-domain (†: statistically significant wrt. Batch APE with $p<0.05$)

From the results of the in-domain experiments with the Autodesk data set it is evident that our proposed online APE method performs not only better than *cdec* and *Thot* (both in batch and online mode) but also better than the strong batch APE method. It achieves significant improvements of 0.54 BLEU, 1.26 TER, and 17.9% precision over the batch APE, which already beats the other online methods. The improvement of our system can be attributed to its ability to learn from the most relevant data and to avoid over-correction by leaving the test segment untouched when no reliable information is found in the knowledge base. As discussed in Section 4.1, several factors like sentence length, sparsity, and translation quality make the WMT2016 data set more challenging to improve for all the online APE methods. In particular, due to the higher translation quality of the *mt* segments, the room for improvement gets lower and the chances of damaging the correct parts are higher. This is visible from the low precision scores reported in Table 3. All the APE methods (batch and online) damage the MT segments in the majority of the cases (precision is lower than 50%). The only exception is our approach that performs significantly better than the batch APE (in terms of TER) and is the only successful method to significantly improve the MT segments in the majority of the cases (61.46%). These experimental results confirm that our proposed online learning APE method based on instance selection to learn only from the most relevant data is sound and reliable.

Building on these results, the main goal of this research is to examine the performance of the online APE methods in a MDTE. This represents a more challenging condition since the system has to adapt to the dynamics of the data processed in a real-time scenario. To emulate this environment, all the online learning methods are trained and tuned on one data set (or domain) and evaluated on the other data set with the possibility to learn from it. In order to capture the peculiarities of the online learning methods over a long run with many data points, we use the training section of the second data set as a test set. The left side of Table 4 reports the performance of all the APE systems when they are trained and tuned on the WMT2016 data set and evaluated on the Autodesk data set. The experimental results reported in the right side of Table 4 are obtained by using the Autodesk data set to train and tune, and the WMT2016 to evaluate. The parameters of our approach (*i.e.* similarity score threshold and minimum number of selected sentence) are the same as computed in the in-domain setting.

	WMT2016 - Autodesk			Autodesk - WMT2016		
	BLEU	TER	Precision (%)	BLEU	TER	Precision (%)
MT	39.91	45.35	N/A	60.90	26.22	N/A
Batch APE	38.09†	46.91†	3.95	55.56†	30.03†	4.03
cdec	38.63†	46.26†	8.36	56.30†	28.98†	7.37
Thot	42.40†	43.45†	58.46	58.11†	28.67†	14.20
Our approach	**43.59**†	**42.44**†	**76.38**	**60.49**†	**26.44**†	**41.37**

Table 4: Performance of the APE systems in a multi-domain translation environment. (†: statistically significant wrt. MT, $p<0.05$; the best scores among the online systems are bold)

In Table 4, the poor performance of the batch APE, which can only leverage the knowledge from the training domain, indicates that the post-editing rules extracted from the training domain are not portable to the test one (even though both datasets belong to IT). This suggests the need of APE approaches that are able to adapt themselves to the incoming data in real-time. Comparing the performance of all the online approaches for both test sets, we notice that our system performs the best with significant gains in all the evaluation metrics. This confirms that our APE system, based on instance selection, is robust enough to work in a MDTE due to its capability to leverage only the most relevant information from a pool of multi-domain segments. Similar to the results on in-domain experiments, significant gains in performance are observed for the Autodesk test set. This does not happen for the WMT2016 test data, for which none of the online APE approaches is able to improve over the MT baseline. For this challenging data set, our approach has the minimal performance degradation (over MT), while the other online systems severely damage the MT segments as confirmed by their low precision (7.37% and 14.20% respectively). One of the common observations, both over the Autodesk and the WMT2016 test sets, is the large difference in precision (17.92% and 27.17% respectively) between the best (our approach) and the second best (Thot) online APE system. This indicates that our approach is more conservative and more suitable to extract and apply domain-specific post-editing rules from a pool of multi-domain data sets, which makes it a more viable and appropriate solution to be deployed in a real-world CAT framework. In the next section, we present some findings on the performance trends of different systems across the entire test set for the *multi-domain* scenario.

6 Performance Analysis

To understand and compare the behavior of different online learning approaches in the long-run, the plot in Figure 2 shows the moving average TER (window of 750 data points) at each segment of the Autodesk test set for the multi-domain experiment (Table 4). As it can be seen, our approach successfully maintains the best performance across the entire test set. As expected, at the beginning of the test set the performance of the online systems is close to the MT system, since there is not much relevant data available to learn from. As time progresses and more segments are processed, a clear trend of performance improvement (with respect to MT) is visible for our method and for the Thot system. This does not hold in the case of cdec, maybe due to the sampling techniques used in the suffix array, which is unable to retrieve relevant samples from the pool of multi-domain data to decode the test segments.

For the WMT2016 test set the moving average TER is shown in Figure 3. As said before, improving translation quality on this test set is more challenging, which is reflected in the graph. Although none of the systems is able to improve over the MT baseline, our system manages to consistently stay close to the MT performance throughout the test set, whereas, all other systems show significant drops. This ensures that our approach is more robust against the domain-shift

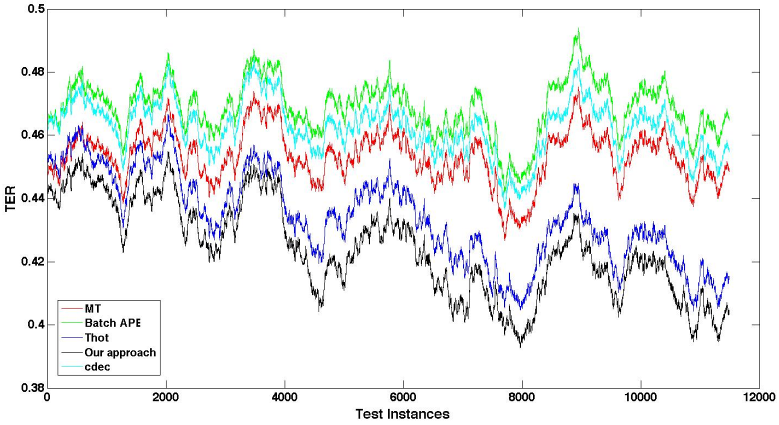

Figure 2: Moving average TER for the Autodesk test set in a *multi-domain* scenario

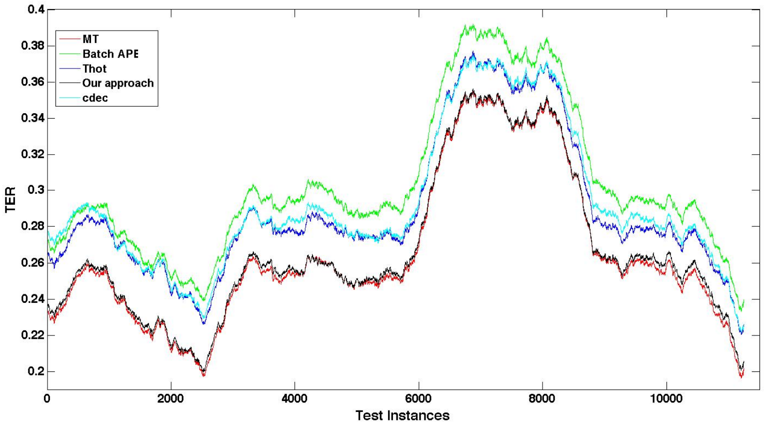

Figure 3: Moving average TER for the WMT2016 test set in a multi-domain scenario

and even in this difficult scenario it is able to maintain stable performance close to the MT without a large deterioration.

To gain further insights about the performance at the segment level, the plot in Figure 4 compares our approach against Thot for the first 300 segments of the Autodesk test set used in the multi-domain experiment. It shows the differences between the segment-level TER of the MT (TER_{MT}) and our approach ($TER_{Our\ approach}$), and MT and Thot (TER_{Thot}) automatically post-edited segments. We notice that our approach modifies less segments compared with Thot, because it builds a model only if it finds relevant data in the knowledge base, otherwise it leaves the MT segment untouched. These untouched MT segments, when modified by Thot, often lead to deterioration rather than to improvements (as seen by many negative peaks for Thot in the Figure 4). This suggests that, compared with the other online approaches, the output obtained with our solution has a higher potential for being useful to human translators. Such usefulness comes not only in terms of a more pleasant post-editing activity, but also in terms of time savings yield by overall better suggestions.

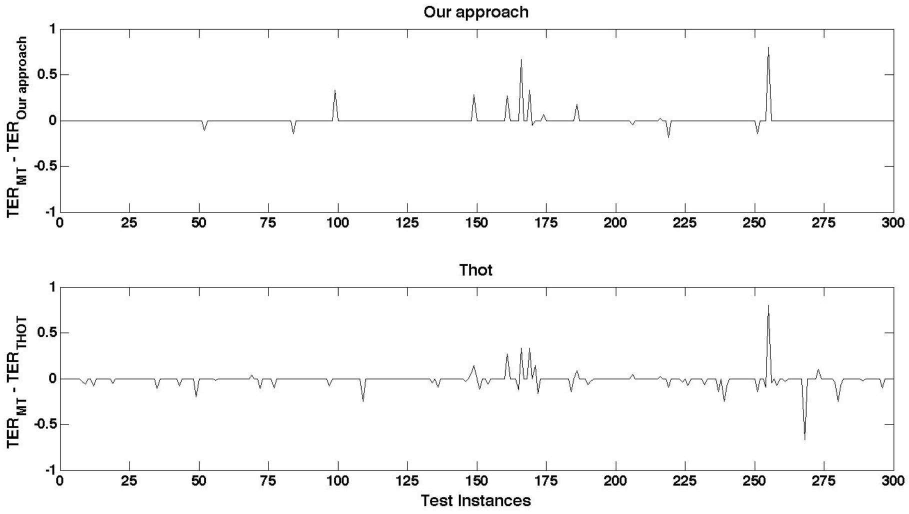

Figure 4: Our approach (top) vs Thot (bottom) performance comparison for the initial test segments (>0 means improvements over the MT, <0 means deterioration of the MT)

7 Conclusion

We addressed the problem of building robust online APE systems in a *multi-domain* translation environment in which the system has to continuously adapt to the dynamics of diverse data processed in real-time. Our evaluation revealed that the online systems that leverage all the available data without considering the peculiarities of each domain are not robust enough to work in a multi-domain translation environment, because they are unable to learn domain-specific post-editing rules. To overcome this limitation, we proposed an online learning framework based on instance selection that has the capability to filter out the most relevant information from a pool of multi-domain data for learning domain-specific post-editing rules. When no reliable information is available our system leaves the MT segments untouched, these segments when automatically post-edited by other systems are often found to get deteriorated. Therefore, the APE suggestions provided by our system to the translators/post-editors are more reliable with better translation quality.

From our experiments in a simulated multi-domain environment, we learn that the post-editing rules are not portable across domains which is revealed by the poor performance of the batch APE system that can leverage only the training data. In the case of online systems that leverage also the test set, it was still a challenging scenario (specially for the Autodesk-WMT2016 data set). Among all the online systems, our proposed approach has the highest improvement on the WMT2016-Autodesk data set, and the least degradation on the Autodesk-WMT2016 data set with respect to the MT quality. Experiments in the *in-domain* setting confirmed that our approach for instance selection is also useful in a single domain scenario. It performed significantly better than the batch APE that already beats cdec and Thot. One common observation from all the experiments in different working scenarios and with different data sets is that our system has the highest precision among all its competitors (MT, batch APE, cdec, and Thot). This indicate that when our system automatically post-edits MT segments, it is more likely to improve the quality of the MT output, which makes it a viable solution to be deployed in a real-word CAT framework.

Acknowledgement

This work has been partially supported by the EC-funded H2020 project QT21 (grant agreement no. 645452), and by the Science Foundation Ireland research grant (no. SFI/12/RC/2289).

References

Béchara, H., Ma, Y., and van Genabith, J. (2011). Statistical post-editing for a statistical mt system. In *Proceedings of the XIII MT Summit*, pages 308–315.

Bertoldi, N., Cettolo, M., and Federico, M. (2013). Cache-based online adaptation for machine translation enhanced computer assisted translation. *Proceedings of the XIV MT Summit*, pages 35–42.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno Yepes, A., Koehn, P., Logacheva, V., Monz, C., Negri, M., Neveol, A., Neves, M., Popel, M., Post, M., Rubino, R., Scarton, C., Specia, L., Turchi, M., Verspoor, K., and Zampieri, M. (2016). Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Scarton, C., Specia, L., and Turchi, M. (2015). Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal.

Cettolo, M., Bertoldi, N., and Federico, M. (2011). Methods for smoothing the optimizer instability in smt. In *Proceedings of the XII MT Summit*, pages 32–39.

Chatterjee, R., C. de Souza, J. G., Negri, M., and Turchi, M. (2016). The fbk participation in the wmt 2016 automatic post-editing shared task. In *Proceedings of the First Conference on Machine Translation*, pages 745–750, Berlin, Germany. Association for Computational Linguistics.

Chatterjee, R., Turchi, M., and Negri, M. (2015a). The fbk participation in the wmt15 automatic post-editing shared task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 210–215.

Chatterjee, R., Weller, M., Negri, M., and Turchi, M. (2015b). Exploring the planet of the apes: a comparative study of state-of-the-art methods for mt automatic post-editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 156–161.

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

Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A. (2011). Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 176–181.

Denkowski, M., Dyer, C., and Lavie, A. (2014). Learning from post-editing: Online model adaptation for statistical machine translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 395–404.

Dugast, L., Senellart, J., and Koehn, P. (2007). Statistical post-editing on systran's rule-based translation system. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 220–223.

Federico, M., Bertoldi, N., and Cettolo, M. (2008). Irstlm: an open source toolkit for handling large scale language models. In *Proceedings of Interspeech*, pages 1618–1621.

Gao, Q. and Vogel, S. (2008). Parallel implementations of word alignment tool. In *Proceedings of Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57.

Hardt, D. and Elming, J. (2010). Incremental re-training for post-editing smt. In *Proceedings of AMTA*.

Hildebrand, A. S., Eck, M., Vogel, S., and Waibel, A. (2005). Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proceedings of EAMT*, pages 133–142.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, pages 388–395.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. System Demonstrations*, pages 177–180.

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

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

Ortiz-Martınez, D. and Casacuberta, F. (2014). The new thot toolkit for fully-automatic and interactive statistical machine translation. In *14th Annual Meeting of the European Association for Computational Linguistics: System Demonstrations*, pages 45–48.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (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.

Parton, K., Habash, N., McKeown, K., Iglesias, G., and de Gispert, A. (2012). Can Automatic Post-Editing Make MT More Meaningful? In *Proceedings of EAMT*, pages 111–118.

Pilevar, A. H. (2011). Using statistical post-editing to improve the output of rule-based machine translation system. *IJCSC*.

Simard, M. and Foster, G. (2013). Pepr: Post-edit propagation using phrase-based statistical machine translation. In *Proceedings of the XIV MT Summit*, pages 191–198.

Simard, M., Goutte, C., and Isabelle, P. (2007a). Statistical Phrase-Based Post-Editing. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 508–515.

Simard, M., Ueffing, N., Isabelle, P., and Kuhn, R. (2007b). Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 203–206.

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

Tatsumi, M. (2009). Correlation between automatic evaluation metric scores, post-editing speed, and some other factors. In *Proceedings of the XII MT Summit*, pages 332–339.

Terumasa, E. (2007). Rule based machine translation combined with statistical post editor for japanese to english patent translation. In *Proceedings of the XI MT Summit*, pages 13–18.

Wuebker, J., Green, S., and DeNero, J. (2015). Hierarchical incremental adaptation for statistical machine translation. In *Proceedings of EMNLP*, pages 1059–1065.

Zhang, Y. and Vogel, S. (2005). An efficient phrase-to-phrase alignment model for arbitrarily long phrase and large corpora. In *Proceedings of EAMT*, pages 294–301.