Passive-Aggressive for On-line Learning in Statistical Machine Translation

Pascual Martínez-Gómez, *Germán Sanchis-Trilles*, *Francisco Casacuberta*
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
{pmartinez,gsanchis,fcn}@dsic.upv.es

Abstract. New variations on the application of the passive-aggressive algorithm to statistical machine translation are developed and compared to previously existing approaches. In online adaptation, the system needs to adapt to real-world changing scenarios, where training and tuning only take place when the system is set-up for the first time. Post-edit information, as described by a given quality measure, is used as valuable feedback within the passive-aggressive framework, adapting the statistical models on-line. First, by modifying the translation model parameters, and alternatively, by adapting the scaling factors present in state-of-the-art SMT systems. Experimental results show improvements in translation quality by allowing the system to learn on a sentence-by-sentence basis.

Keywords: on-line learning, passive-aggressive, statistical machine translation

1 Introduction

Online passive-aggressive (PA) algorithms [4] are a family of margin-based online learning algorithms that are specially suitable for adaptation tasks where a convenient change in the value of the parameters of our models is desired after every sample is presented to the system. The general idea is to learn a weight vector representing a hyperplane such that differences in quality also correspond to differences in the margin of the instances to the hyperplane. The update is performed in a characteristic way by trying to achieve at least a unit margin on the most recent example while remaining as close as possible to the current hyperplane.

Different ways to apply the PA framework to statistical machine translation (SMT) are analised. SMT systems use mathematical models to describe the translation task and to estimate the probability of translating a source sentence $\mathbf{x}$ into a target sentence $\mathbf{y}$. Recently, a direct modelling of the posterior probability $\Pr(\mathbf{x} \mid \mathbf{y})$ has been widely adopted. To this purpose, different authors [11, 8] propose the use of the so-called log-linear models, where the decision rule is given by the expression

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \sum_{m=1}^{M} \lambda_m h_m(\mathbf{x}, \mathbf{y}) = \operatorname*{argmax}_{\mathbf{y}} \boldsymbol{\lambda} \mathbf{h}(\mathbf{x}, \mathbf{y}) = \operatorname*{argmax}_{\mathbf{y}} s(\mathbf{x}, \mathbf{y}) \qquad (1)$$

where $h_m(\mathbf{x}, \mathbf{y})$ is a score function representing an important feature for the translation of $\mathbf{x}$ into $\mathbf{y}$, M is the number of models (or features) and λ_m are the weights of the log-linear combination. $s(\mathbf{x}, \mathbf{y})$ is a score representing how good $\mathbf{x}$ translates into $\mathbf{y}$. Common feature functions $h_m(\mathbf{x}, \mathbf{y})$ include different translation models (TM), but also distortion models or even the target language model. $\mathbf{h}(\cdot|\cdot)$ and $\boldsymbol{\lambda}$ are estimated by means of training and development sets, respectively.

In order to capture context information, *phrase-based* models [16] were introduced, widely outperforming single word models [3]. The main idea is to segment source sentence $\mathbf{x}$ into *phrases* (i.e. word sequences), and then to translate each source phrase $\tilde{x}_k \in \mathbf{x}$ into a target phrase $\tilde{y}_k$. Those models were employed throughout this work.

Adjusting both $\mathbf{h}$ or $\boldsymbol{\lambda}$ leads to an important problem in SMT: whenever the text to be translated belongs to a different domain than the training corpora, translation quality diminishes significantly [3]. For this reason, the problem of *adaptation* is very common in SMT, where the objective is to improve the performance of systems trained and tuned on out-of-domain data by using very limited amounts of in-domain data.

Adapting a system to changing tasks is specially interesting in the Computer Assisted Translation (CAT) [2] and Interactive Machine Translation (IMT) paradigms [1], where the collaboration of a human translator is essential to ensure high quality results. Here, the SMT system proposes a hypothesis to a human translator, who may amend the hypothesis to obtain an acceptable translation, and after that expects the system to learn from its own errors, so that it is not necessary to correct the same error again. The challenge is then to make the best use of every correction provided by adapting the system *online*, i.e. without performing a complete retraining which is too costly.

In this work, the performance of PA with two adaptation strategies is analysed, namely feature vector and scaling factor adaptation, with the purpose of using feedback information to improve subsequent translations in a sentence-by-sentence basis.

Similar work is briefly detailed in the following Section. PA algorithms are reviewed in Section 3. Their application to SMT is described in Section 4. Experiments conducted are analysed in Section 5, and conclusions and future work are listed in Section 6.

2 Related Work

In [10], an online learning application is presented for IMT, incrementally updating model parameters by means of an incremental version of the Expectation-Maximisation algorithm and allowing for the inclusion of new phrase pairs. We propose the use of a dynamic reranking algorithm which is applied to a $nbest$ list, regardless of its origin. In addition, in [10], only $\mathbf{h}$ is adapted, whereas here we also analyse the adaptation of $\boldsymbol{\lambda}$.

In [13] the authors propose the use of the PA framework [4] for updating the feature functions $\mathbf{h}$. The obtained improvements were very limited, since adapting $\mathbf{h}$ is a very sparse problem. Hence, in the present paper, the adaptation of the $\boldsymbol{\lambda}$ will be compared to the adaptation of $\mathbf{h}$, which is shown in [14] to be a good adaptation strategy. In [14], the authors propose the use of a Bayesian learning technique in order to adapt the scaling factors based on an adaptation set. In contrast, our purpose is to perform online adaptation, i.e. to adapt system parameters after each new sample has been provided.

Another difference between [13] and the present work is that they propose to model the user feedback by means of BLEU score [12], which is quite commonly used in SMT. Such score measures precision of n-grams with a penalty for sentences that are too short. However, BLEU is not well defined on the sentence level, since it implements a geometric average which is zero whenever no common 4-gram exists between reference and hypothesis. In the present work, we propose the use of TER [15] instead. TER is similar to the word error rate criterion of speech recognition, but allowing shifts of word sequences. TER is well defined on the sentence level, and, furthermore, in [15] it is shown to correlate better with human judgement.

3 The passive-aggressive algorithm

PA [4] is a family of margin-based, on-line learning algorithms that update model parameters after each observation has been seen. In this case, PA is applied to a regression problem, where target value $\hat{\mu}(\mathbf{y})_t \in \mathbb{R}$ has to be predicted by the system for input observation $\mathbf{x}_t \in \mathbb{R}^n$ at time t by using a linear regression function $\hat{\mu}(\mathbf{y})_t = \mathbf{w}_t \cdot \mathbf{x}_t$.

After every prediction, the true target value $\mu(\mathbf{y})_t \in \mathbb{R}$ is received and the system suffers an instantaneous loss according to a sensitivity parameter ϵ:

$$l_\epsilon(\mathbf{w}; (\mathbf{x}, \mu(\mathbf{y}))) = \begin{cases} 0 & \text{if } |\mathbf{w} \cdot \mathbf{x} - \mu(\mathbf{y})| \leq \epsilon \\ |\mathbf{w} \cdot \mathbf{x} - \mu(\mathbf{y})| - \epsilon & \text{otherwise} \end{cases} \tag{2}$$

If the system's error falls below ϵ, the loss suffered by the system is zero and the algorithm remains *passive*, that is, $\mathbf{w}_{t+1} = \mathbf{w}_t$. Otherwise, the loss grows linearly with the error $|\hat{\mu}(\mathbf{y}) - \mu(\mathbf{y})|$ and the algorithm *aggressively* forces an update of the parameters.

The idea behind the PA algorithm is to modify the parameter values of the regression function so that it achieves a zero loss function on the current observation $\mathbf{x}_t$, while remaining as close as possible to the previous weight vector $\mathbf{w}_t$. That is, formulated as an optimisation problem subject to a constraint [4]:

$$\mathbf{w}_{t+1} = \underset{\mathbf{w} \in \mathbb{R}^n}{\text{argmin}} \frac{1}{2} ||\mathbf{w} - \mathbf{w}_t||^2 + C\xi^2 \qquad \text{s.t.} \quad l_\epsilon(\mathbf{w}; (\mathbf{x}, \mu(\mathbf{y}))) = 0 \tag{3}$$

where ξ^2 is, according to the so-called PA Type-II, a squared slack variable scaled by the aggressivity factor C. As in classification tasks, it is common to add a slack variable into the optimisation problem to get more flexibility during the learning process.

It is only left to add the constraint together with a Lagrangian variable and set the partial derivatives to zero to obtain the closed form of the update term. In Section 4, the update term for every adaptation strategy ($\boldsymbol{\nu}_t$ and $\hat{\boldsymbol{\lambda}}_t$) is detailed.

4 Passive-aggressive in SMT

4.1 Feature vector adaptation

As described in [13], PA can be used for adapting the translation scores within state-of-the-art TMs. First, we need to define $h_{TM}(\mathbf{x}, \mathbf{y})$ as the combination of n TMs implicit in current translation systems, which are typically specified for all phrase pairs $(\tilde{x}_k, \tilde{y}_k)$:

$$h_{TM}(\mathbf{x}, \mathbf{y}) = \sum_n \lambda_n \sum_k h_n(\tilde{x}_k, \tilde{y}_k) \tag{4}$$

where h_{TM} can be considered as a single feature function h in Eq. 1. Then, we can study the effect of adapting the TMs in an online manner by adapting h_{TM}. Although there might be some reasons for adapting all the score functions $\mathbf{h}$, in the present paper we focus on analysing the effect of adapting only the TMs. By considering $\forall m \notin TM$: $h^t_m(\cdot,\cdot) = h_m(\cdot,\cdot)$, and defining an auxiliary function $\mathbf{u}_t(\mathbf{x},\mathbf{y})$ such that

$$h^t_{TM}(\mathbf{x},\mathbf{y}) = \sum_n \lambda_n \sum_k u_t(\tilde{x}_k,\tilde{y}_k) h_n(\tilde{x}_k,\tilde{y}_k) = \mathbf{u}_t(\mathbf{x},\mathbf{y})\mathbf{h}_{TM}(\mathbf{x},\mathbf{y}),$$

the decision rule in Eq. 1 is approximated, by only adapting $h_{TM}(\cdot|\cdot)$, as

$$\hat{\mathbf{y}}_t = \operatorname*{argmax}_{\mathbf{y}} \sum_{m \neq TM} \lambda_m h_m(\mathbf{x},\mathbf{y}) + h^t_{TM}(\mathbf{x},\mathbf{y}). \tag{5}$$

Let $\mathbf{y}$ be the hypothesis proposed by the system, and $\mathbf{y}^*$ the best hypothesis the system is able to produce in terms of translation quality (i.e. the most similar sentence with respect to reference translation proposed by the user $\mathbf{y}^\tau$). Ideally, we would like to adapt the model parameters (be it $\boldsymbol{\lambda}$ or $\mathbf{h}$) so that $\mathbf{y}^*$ is rewarded.

We define the difference (or loss) in translation quality between the proposed hypothesis $\mathbf{y}$ and the best hypothesis $\mathbf{y}^*$ in terms of a given quality measure $\mu(\cdot)$:

$$l(\mathbf{y}) = |\mu(\mathbf{y}^\tau,\mathbf{y}) - \mu(\mathbf{y}^\tau,\mathbf{y}^*)|, \tag{6}$$

where the absolute value has been introduced in order to preserve generality, since in SMT some of the quality measures used, such as TER [15], represent an error rate (i.e. the lower the better), whereas others such as BLEU [12] measure precision (i.e. the higher the better). The difference in probability between $\mathbf{y}$ and $\mathbf{y}^*$ is proportional to

$$\phi(\mathbf{y}) = s(\mathbf{x},\mathbf{y}^*) - s(\mathbf{x},\mathbf{y}). \tag{7}$$

Ideally, we would like that increases or decreases in $l(\cdot)$ correspond to increases or decreases in $\phi(\cdot)$, respectively: if a candidate hypothesis $\mathbf{y}$ has a translation quality $\mu(\mathbf{y})$ which is very similar to the translation quality provided by $\mu(\mathbf{y}^*)$, we would like that such fact is reflected in the translation score s, i.e. $s(\mathbf{x},\mathbf{y})$ is very similar to $s(\mathbf{x},\mathbf{y}^*)$. The purpose of our online procedure should be to promote such correspondence after processing sample t. The update step for $\mathbf{u}_t(\mathbf{x},\mathbf{y})$ can be defined as $\mathbf{u}_{t+1}(\mathbf{x},\mathbf{y}) = \mathbf{u}_t(\mathbf{x},\mathbf{y}) + \boldsymbol{\nu}_t$, where $\mathbf{u}_t(\mathbf{x},\mathbf{y})$ is the update function learnt after observing the previous $(\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_{t-1},\mathbf{y}_{t-1})$ pairs, and $\boldsymbol{\nu}_t$ is the solution to the optimisation problem

$$\min_{\mathbf{u},\xi>0} \left(\frac{1}{2}||\mathbf{u}-\mathbf{u}_t||^2 + C\xi^2\right) \tag{8}$$

subject to constraint $\mathbf{u}_t(\mathbf{x},\mathbf{y})\mathit{\Phi}_t(\mathbf{y}) \geq \sqrt{l(\mathbf{y})} - \xi$, with $\mathit{\Phi}_t(\mathbf{y}) = [\phi(\tilde{y}_1),\ldots,\phi(\tilde{y}_K)]' \approx \mathbf{h}_{TM}(\mathbf{x},\mathbf{y}^*) - \mathbf{h}_{TM}(\mathbf{x},\mathbf{y})$, since all the rest of score functions except $\mathbf{h}_{TM}$ remain constant and the only feature functions we intend to adapt are $\mathbf{h}_{TM}$. Then, the solution to Equation 8 according to PA Type-II has the form [13]:

$$\boldsymbol{\nu}_t = \mathit{\Phi}_t(\mathbf{y}) \frac{\sqrt{l_t(\mathbf{y})} - \mathbf{u}_t(\mathbf{x},\mathbf{y})\mathit{\Phi}_t(\mathbf{y})}{||\mathit{\Phi}_t(\mathbf{y})||^2 + \frac{1}{C}} \tag{9}$$

In [13], the update is triggered only when the proposed hypothesis violates the constraint $\mathbf{u}_t(\mathbf{x},\mathbf{y})\mathit{\Phi}_t \geq \sqrt{l_t(\mathbf{y})}$.

4.2 Scaling factor adaptation

A coarse-grained technique for tackling with the online learning problem in SMT implies adapting the log-linear weights $\boldsymbol{\lambda}$. After the system has received the sentence $\mathbf{y}_t^{\mathcal{T}}$ as correct reference for an input sentence $\mathbf{x}_t$, the idea is to compute the best weight vector $\hat{\boldsymbol{\lambda}}_t$ corresponding to the sentence pair observed at time t. Once $\hat{\boldsymbol{\lambda}}_t$ has been computed, $\boldsymbol{\lambda}_t$ can be updated towards a new weight vector $\boldsymbol{\lambda}_{t+1}$, for a certain learning rate α, as:

$$\boldsymbol{\lambda}_{t+1} = (1-\alpha)\boldsymbol{\lambda}_t + \alpha\hat{\boldsymbol{\lambda}}_t \tag{10}$$

As done with $\boldsymbol{\nu}_t$ in Section 4.1, the update term for computing $\boldsymbol{\lambda}_{t+1}$ is given by

$$\hat{\boldsymbol{\lambda}}_t = \varPhi_t(\mathbf{y})\frac{\sqrt{l_t(\mathbf{y})} - \boldsymbol{\lambda}_t\varPhi_t(\mathbf{y})}{||\varPhi_t(\mathbf{y})||^2 + \frac{1}{C}}, \tag{11}$$

where $\varPhi_t(\mathbf{y}) = [\phi_1(\mathbf{y}), \ldots, \phi_M(\mathbf{y})]' = \mathbf{h}(\mathbf{x}, \mathbf{y}^*) - \mathbf{h}(\mathbf{x}, \mathbf{y})$, including all feature functions. An update is triggered only when constraint $\boldsymbol{\lambda}_t\varPhi_t(\mathbf{y}) \geq \sqrt{l_t(\mathbf{y})}$ is violated.

4.3 Heuristic variations

Several update conditions different to the ones described above have been explored in this paper. The most obvious is to think that an update has to be performed every time that the quality of a predicted hypothesis $\mathbf{y}$ is lower than the best possible hypothesis $\mathbf{y}_t^*$ in terms of a given quality measure μ. That is, when $\exists\mathbf{y}^* : |\mu(\mathbf{y}_t, \mathbf{y}^*) - \mu(\mathbf{y}_t, \mathbf{y})| > 0$.

In feature vector adaptation, the key idea is to reward those phrases that appear in $\mathbf{y}^*$ but did not appear in $\mathbf{y}$, and, symmetrically, to penalise phrases that appeared in $\mathbf{y}$ but not in $\mathbf{y}^*$. When adapting $\boldsymbol{\lambda}$, the idea is to adjust the discriminative power of models by means of shifting the value of their scaling factors towards the desired value.

5 Experiments

5.1 Experimental setup

Given that a true CAT scenario is very expensive for experimentation purposes, since it requires a human translator to correct every hypothesis, we will be simulating such scenario by using the reference present in the test set. However, such reference will be fed one at a time, given that this would be the case in an online CAT process.

Translation quality will be assessed by means of BLEU and TER scores. It must be noted that BLEU measures precision, i.e. the higher the better, whereas TER is an error rate, i.e. the lower the better. As mentioned in Section 2, BLEU may often be zero for all hypotheses, which means that $\mathbf{y}^*$ is not always well defined and it may not be possible to compute it. Such samples will not be considered within the online procedure.

As baseline system, we trained a SMT system on the Europarl [6] training data, in the partition established in the Workshop on SMT of the NAACL 2009[1]. Since our purpose is to analyse the performance of the PA algorithm in an online adaptation scenario, we also considered the use of the News Commentary (NC) test set of the 2009 ACL

[1] http://www.statmt.org/wmt09/

Table 1. Characteristics of the Europarl corpus and NC09 test set. OoV stands for "Out of Vocabulary" words, K for thousands of elements and M for millions of elements.

		Es	En	Fr	En	De	En
Training	Sentences	1.3M		1.2M		1.3M	
	Run. words	27.5M	26.6M	28.2M	25.6M	24.9M	26.2M
	Vocabulary	125.8K	82.6K	101.3K	81.0K	264.9K	82.4K
Development	Sentences	2000		2000		2000	
	Run. words	60.6K	58.7K	67.3K	48.7K	55.1K	58.7K
	OoV. words	164	99	99	104	348	103
NC 09 test	Sentences	2525		2051		2051	
	Run. words	68.1K	65.6K	72.7K	65.6K	62.7K	65.6K
	OoV. words	1358	1229	1449	1247	2410	1247

shared task on SMT. Statistics are provided in Table 1. The open-source MT toolkit Moses [7] was used in its default setup, and the 14 weights of the log-linear combination were estimated using MERT [9] on the Europarl development set. Additionally, an interpolated 5-gram language model and Kneser-Ney smoothing [5] was estimated.

Experiments were performed on the English–Spanish, English–German and English–French language pairs, in both directions and for NC test sets of 2008 and 2009. However, in this paper only the results for English $\rightarrow$ French are presented, for space reasons. In addition, we only report results for the 2009 test set. Nevertheless, the results presented here were found to be coherent in all experiments conducted.

As for the different parameters adjustable in the algorithms described in Section 4.1 and 4.2, they were set according to preliminary investigation to $C \rightarrow \infty$ ($\frac{1}{C} = 0$ was used) in both approaches and $\alpha = 0.01$ in scaling factor adaptation. Instead of using the true best hypothesis, the best hypothesis within a given $nbest(\mathbf{x})$ list was selected.

5.2 Experimental results

We analysed the performance of the different PA variations described in Section 4, both in terms of BLEU and in terms of TER, and both for adapting $\mathbf{h}$ and $\boldsymbol{\lambda}$. Results for varying order of $nbest$ can be seen in Fig. 1. Although the final scores are reported for the whole test set, all experiments described here were performed following an online CAT approach: each reference sentence was used for adapting the system parameters after such sentence has been translated and its translation quality has been assessed.

It can be seen that the heuristic PA variation yields a small improvement when optimising TER. However, such improvement is not mirrored when optimising BLEU, and hence we assume it is not significant. It can also be seen that adapting $\boldsymbol{\lambda}$ leads to consistently better performance than adapting $\mathbf{h}$. Although adapting $\mathbf{h}$ provides much more flexibility, we understand that adapting $\mathbf{h}$ is a very sparse problem.

The techniques analysed perform much better in terms of TER than in terms of BLEU. Again, it is worth remembering that BLEU is not well defined at the sentence level, and hence the fact that PA has more trouble using it was expected.

In Fig. 2, the evolution of TER throughout the whole test set is plotted for the adaptation of $\mathbf{h}$ and $\boldsymbol{\lambda}$ when setting the size of the $nbest$ list to 1000. In this figure, average TER scores up to the t-th sentence is considered. The reason for plotting average TER is that plotting individual sentence TER scores would result in a very chaotic, unreadable plot, as it can still be seen in the first 100 sentences. Again, in this Figure it also emerges that adapting $\boldsymbol{\lambda}$ leads to much larger improvements than adapting $\mathbf{h}$.

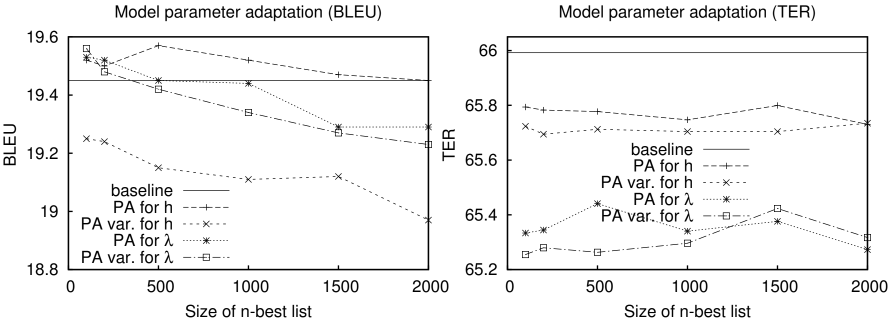

Fig. 1. Final BLEU and TER scores for the NC 2009 test set, English → French when adapting feature functions $\mathbf{h}$ and when adapting scaling factors $\boldsymbol{\lambda}$. `PA` stands for PA as described in Section 4 and `PA var.` for the heuristic variation described in Section 4.3.

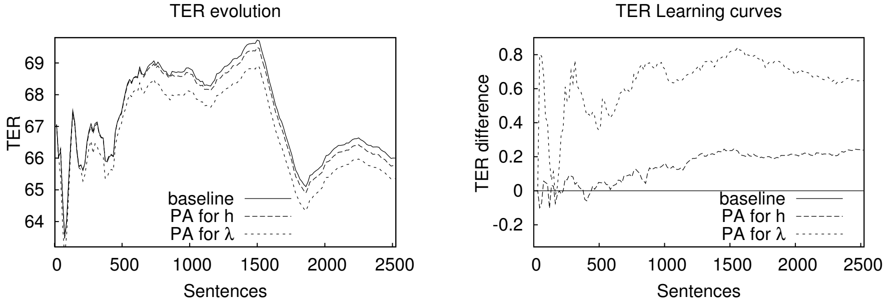

Fig. 2. TER evolution and learning curves when adapting feature functions $\mathbf{h}$ and scaling factors $\boldsymbol{\lambda}$, considering all 2525 sentences within the NC 2009 test set. So that the plots are clearly distinguishable, only 1 every 15 points has been drawn.

Although it appears that the learning curves peak at about 1500 sentences, this finding is not coherent throughout all experiments carried out, since such peak ranges from 300 to 2000 in other cases. This means that the particular shape of the learning curves depends strongly on the chosen test set, and that the information that can be extracted is only whether or not the algorithms implemented provide improvements.

One last consideration involves computation time. When adapting $\boldsymbol{\lambda}$, implemented procedures take about 100 seconds to rerank the complete test set, whereas in the case of adapting $\mathbf{h}$ the time is about 25 minutes. We consider this fact important since in a CAT scenario the user is waiting actively for the system to produce a hypothesis.

6 Conclusions and future work

The passive-aggressive algorithm has been analysed for its application in an online scenario, adapting system parameters after each observation. Feedback information has been included into an SMT system, increasing the perception of its own performance.

The passive-aggressive algorithm and a proposed heuristic variation have been applied to two tasks with different characteristics. Feature function adaptation is a sparse problem in the order of thousands of parameters that need to be adapted, whereas the scaling factor adaptation only has around 14 parameters to adapt. This might be one of the reasons for the passive-aggressive algorithm to perform better in the latter task.

Two quality scores have also been used during the experiments and the behaviour of the system allows us to extract one more conclusion. When optimising BLEU, the performance of the algorithm is consistently lower than when optimising TER. We believe that the reason for this is that BLEU is not well defined at the sentence level.

In future work, it would be interesting to observe the impact of smoothed quality scores on the performance of the algorithms.

Acknowledgements

This paper is based upon work supported by the EC (FEDER/FSE) and the Spanish MICINN under projects MIPRCV "Consolider Ingenio 2010" (CSD2007-00018) and iTrans2 (TIN2009-14511). Also supported by the Spanish MITyC under the erudito.com (TSI-020110-2009-439) project, by the Generalitat Valenciana under grant Prometeo/2009/014 and scholarship GV/2010/067 and by the UPV under grant 20091027.

References

1. Barrachina, S., et al.: Statistical approaches to computer-assisted translation. Computational Linguistics 35(1), 3–28 (2009)
2. Callison-Burch, C., Bannard, C., Schroeder, J.: Improving statistical translation through editing. In: Proc. of 9th EAMT workshop ”Broadening horizons of machine translation and its applications”. Malta (April 2004)
3. Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., Schroeder, J.: (meta-) evaluation of machine translation. In: Proc. of the Workshop on SMT. pp. 136–158. ACL (June 2007)
4. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. Journal of Machine Learning Research 7, 551–585 (2006)
5. Kneser, R., Ney, H.: Improved backing-off for m-gram language modeling. IEEE Int. Conf. on Acoustics, Speech and Signal Processing II, 181–184 (May 1995)
6. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: Proc. of the MT Summit X. pp. 79–86 (2005)
7. Koehn et al., P.: Moses: Open source toolkit for statistical machine translation. In: Proc. of the ACL Demo and Poster Sessions. pp. 177–180. Prague, Czech Republic (2007)
8. Och, F., Ney, H.: Discriminative training and maximum entropy models for statistical machine translation. In: Proc. of the ACL'02. pp. 295–302 (2002)
9. Och, F.: Minimum error rate training for statistical machine translation. In: Proc. of ACL'03. pp. 160–167 (2003)
10. Ortiz-Martínez, D., García-Varea, I., Casacuberta, F.: Online learning for interactive statistical machine translation. In: Proceedings of NAACL HLT. Los Angeles (Jun 2010)
11. Papineni, K., Roukos, S., Ward, T.: Maximum likelihood and discriminative training of direct translation models. In: Proc. of ICASSP'98. pp. 189–192 (1998)
12. Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: A method for automatic evaluation of machine translation. In: Proc. of ACL'02 (2002)
13. Reverberi, G., Szedmak, S., Cesa-Bianchi, N., et al.: Deliverable of package 4: Online learning algorithms for computer-assisted translation (2008)
14. Sanchis-Trilles, G., Casacuberta, F.: Log-linear weight optimisation via bayesian adaptation in statistical machine translation. In: Proc. of COLING'10. Beijing, China (August 2010)
15. Snover, M., et al.: A study of translation edit rate with targeted human annotation. In: Proc. of AMTA'06. Cambridge, Massachusetts, USA (August 2006)
16. Zens, R., Och, F., Ney, H.: Phrase-based statistical machine translation. In: Proc. of KI'02. pp. 18–32 (2002)