Evaluating the Learning Curve of Domain Adaptive Statistical Machine Translation Systems

Nicola Bertoldi Mauro Cettolo Marcello Federico
Fondazione Bruno Kessler
via Sommarive 18
38123 Trento, Italy
<surename>@fbk.eu

Christian Buck
University of Edinburgh
10 Crichton Street
EH8 9AB Edinburgh, UK
christian.buck@ed.ac.uk

Abstract

The new frontier of computer assisted translation technology is the effective integration of statistical MT within the translation workflow. In this respect, the SMT ability of incrementally learning from the translations produced by users plays a central role. A still open problem is the evaluation of SMT systems that evolve over time. In this paper, we propose a new metric for assessing the quality of an adaptive MT component that is derived from the theory of learning curves: the percentage slope.

1 Introduction

Translation memories and computer assisted translation (CAT) tools are currently the dominant technologies in the translation and localization market, but recent achievements in statistical MT have raised new expectations in the translation industry. So far, statistical MT has focused on providing ready-to-use translations, rather than outputs that minimize the effort of a human translator. The MateCAT project[1] aims at pushing what can be considered the new frontier of CAT technology: how to effectively integrate statistical MT within the translation workflow.

One pursued research direction is developing *domain adaptive* SMT models, i.e. models that dynamically adapt to the translations that are continuously added to the translation memory by the user during her/his work. The ideal goal is to progressively reduce the mismatch between training and testing data, in such a way that the adapted SMT engine will be able to provide the user with *useful suggestions* – i.e. perfect or worth being post-edited – when the translation memory fails to retrieve perfect or almost perfect matches. Among the well known machine learning paradigms that fit with this scenario are *online learning* and *incremental learning*, which basically differ in the amount of data that is employed to dynamically adapt the system: a single piece of data in the first case and a batch of data in the latter. Notice that in both cases one assumes that domain adaptation is performed efficiently, i.e. by only processing the newly received data. Moreover, although the quantity of acquired in-domain data is generally limited, their high quality and relevance to the translation task justify their exploitation by all means possible.

Domain adaptive SMT embeds two challenges: (1) the design of effective adaptation algorithms, and (2) the evaluation of MT systems evolving over time. Since the ultimate goal of our efforts is to increase the productivity of human translators, the most accurate assessment methodology would be of course to run a field test. This way, we could compare productivity of human translators receiving suggestions from an MT engine featuring dynamic domain adaptation against the productivity of human translators working with a static MT engine. As this evaluation is infeasible during daily MT development, we can resort to the several automatic MT metrics, which however, as we will see later, are unsuitable to track the dynamic behaviors we are interested to investigate. Metrics for measuring performance in the case of interactive MT, see for example (Khadivi,

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

Proceedings of the 7th Workshop on Statistical Machine Translation, pages 433–441,
Montréal, Canada, June 7-8, 2012.

2008), like Key-Stroke Ratio (KSR), Mouse-Action Ratio (MAR), Key-Stroke and Mouse-Action Ratio (KSMR) are known to correlate well with the productivity of human translators, but their computation requires the actual use of an interactive MT system, i.e. a field test.

In the SMART project,[2] the evaluation of adaptive interactive MT is explored (Cesa-Bianchi et al., 2008). While no specific metric is proposed, the analysis is based on a plot of cumulative differences of BLEU scores between a baseline and an adaptive system. These differences are computed sentence by sentence and present an interesting view of the dynamic change of the MT system. We are going to further elaborate on this idea.

Other metrics like Character Error Rate (CER) and Translation Edit Rate (TER) would accurately predict the translators' productivity if references were generated by using the CAT system; on the contrary, references are usually, as in this paper, generated from scratch based only on the source text and can thus be quite far from CAT-based translations, both lexically and syntactically. The Human-targeted variant of TER, HTER (Snover et al., 2006), needs human intervention and is therefore unfit to meet our requirements.

The main goal of this paper is to design an objective automatic evaluation methodology for an MT system adapting over time. We propose to use the *percentage slope* from the theory on learning curves to measure the learning ability of adaptive MT systems.

To assess the proposed metric, we have implemented a simple but effective adaptation strategy suitable for an MT system integrated in a CAT tool. We show that the percentage slope is able to expose different dynamic behaviors, such as learning, no learning, and forgetting.

2 Dynamic Adaptation Framework

In the MateCAT project scenario, the MT system, which is embedded in the CAT tool to increase the translators' productivity, adapts over time by exploiting translations generated by the user. The adapted system is then used to provide the user with translation suggestions for the next sentences. We refer to this process as *dynamic* (or *incremental*) *adaptation* to emphasize that adaptation happens continuously based on a stream of data.

2.1 Abstract View of the Adaptation Process

From an abstract point of view, the framework of incremental adaptation can be summarized as follows:

i) before the process starts, an initial system is built on available data including a parallel corpus;
ii) a stream of parallel data becomes available that is split into blocks of (not necessarily) similar size;
iii) the first/next block is considered, but only the source is available yet;
iv) the latest instance of the adapting system translates the source text of the current block;
v) the target part of the current block becomes available for use;[3]
vi) the system is adapted using the current parallel block and possibly all the previous ones;
vii) the loop continues from step iii) until all blocks are processed.

In each adaptation step, all of the data available so far can be used, but no look ahead is possible. Note that, in principle, each block is translated with a different instance of the adapting system; hence, the same text occurring in two different blocks can be translated differently.

2.2 Evaluation Goals and Requirements

Although dynamic adaptation is closely related to static domain adaptation (Foster and Kuhn, 2007), in this scenario we are not interested in the quality of the final model. In fact, this model is only available once the stream is depleted and therefore is not used anymore.

What we are interested in, and what we want to compare among different approaches, is the systems evolvement over time.

Consider a translator who uses such an incrementally adapting system and performs post-editing on its suggested translations. The highest productivity

[2] http://www.smart-project.eu

[3] In the CAT framework, the target part of a block is the translation post-edited by the user.

gain is achieved when the adaptation is quick and persistent.

Even though in this paper we are concerned with an automatic metric, it is important to keep the use case of CAT in mind, in particular the presence of a human translator. The TransType2 project[4] has found that repeated correction of the same error is strongly disliked by editors (Macklovitch, 2006) and may lead to rejection of the entire system. Similarly, segments that were translated correctly by previous, less adapted systems, should not be negatively affected by updates. We will refer to these particular aspects of adaptation as *backward reliability*.

Automatic measures, which are aimed at static MT modules, can not take the evolution of the system into account and are therefore unable to pinpoint such problems. Thus, they are not suitable for the dynamic adaptation scenario.

A new evaluation methodology should satisfy the following requirements:

- ability to compare different strategies
- show behavior over time and reward early improvements and consistent adaptation
- expose possible overfitting, i.e. check whether generalization is lost due to overly aggressive adaptation
- strong correlation to human productivity
- estimate benefit over a static baseline model without adaptation
- check backward reliability.

2.3 Evaluation Protocol

The performance of adaptive systems as sketched in Section 2.1 is evaluated on different parts of the stream as opposed to the global evaluation used for static systems. We distinguish between two protocols which differ in their use of historic data.

For *block-wise evaluation* only the translations of the most recent block are evaluated with respect to the correct translations once these become available. Any static automatic MT score, e.g. TER (Snover et al., 2006), BLEU (Papineni et al., 2001), can be used, provided that it is reliable on a block of usually relatively small size.

In contrast, in *incremental evaluation* the scores are computed on all blocks available so far. The translations of previous blocks are kept fixed, i.e. blocks are *not* translated again once a newly adapted system becomes available as this new system has already seen this data.

Both the block-wise and incremental protocols yield a sequence of scores that reflects the adaptation behavior over time. The former is useful to expose potential weaknesses as discussed above: we expect to see improvement at first and after a while, when enough adaptation data is available, a level curve. If this is not the case, this indicates a problem:

i) should the scores deteriorate over time we might be facing overfitting, possibly due to unexpected heterogeneity in our corpus;

ii) if the scores continue to improve, then the adaptation method is not aggressive enough and the system underfits.

The incremental evaluation on the other hand allows for easy comparison of different adaptation strategies. While the performance on the most recent block becomes less important over time, the performance on all the blocks processed so far nicely reflects the utility of the system in the application setting.

The metric we are going to propose in the next section processes such sequences of partial scores. It accumulates the trend into a single number and offers an interpretation that relates adaptive behavior to productivity gains.

3 The Percentage Slope

Learning curves (see (Stump P.E., 2002) for a detailed introduction) are mathematical models used to estimate the efficiency gain when an activity is repeated. The *learning effect* was noted in industrial environment: the underlying notion is that when people repeat an activity, there tends to be a gain in efficiency. That is exactly the expected behavior of our dynamically adapting MT system: it should improve its performance on texts including terms and expressions whose proper translation has been previously provided. Thus we decided to exploit elements from learning theory to measure the evolution of translation capability.

Several learning curve models have been proposed, but only two are in widespread use, the *unit*

[4] http://tt2.atosorigin.es

(U) model due to Crawford and the *cumulative average* (CA) model due to Wright. Both models are based on a common mathematical form:

$$y = ax^b \quad (1)$$

where:

- a represents the theoretical labor hours required to build the first unit produced (a positive number)
- b represents the rate of learning (negative value, except for "forgetting")
- x represents the number of an item in the production sequence (unit $\#1, \#2, \#3, \ldots$)

The models differ in the interpretation of y:

`U`: y is the labor hours required to build unit $\#x$

`CA`: y is the average labor hours per unit required to build the first x units

Since b is a mathematically appropriate but counter-intuitive number for describing the slope, the *percentage slope* S is typically used:

$$S = 10^{b \log_{10}(2)+2} \quad (2)$$

S provides the rate of learning on a scale of 0 to 100, as a percentage. A 100% slope represents no learning at all, zero percentage reflects a theoretically infinite rate of learning. In practice, human operations hardly ever achieve a rate of learning faster than 70% as measured on this scale.

The correspondence between our block-wise evaluation (Section 2.3) with the U model, and the incremental evaluation with the CA model is straightforward. In the first case, y is the number of errors done in the translation of the block $\#x$; in the second case, y is the average number of errors (that is the TER score or the 100-BLEU score) made on the first x blocks.

From a practical point of view, the sequence of scores can be provided while the adapting system is being used; the learning curve which best matches the sequence is then found[5] and eventually the percentage slope S is computed.

[5]Notice that the best fitting learning curve can be estimated in the log scale with a simple linear regression analysis.

set	#sent.	#src words	#tgt words
train	1.2M	18.9M	19.4M
test	3.4k	57.0k	61.4k

Table 1: Overall statistics on parallel data of the IT domain used for training and testing the SMT system. Counts of (English) source words and (Italian) target words refer to tokenized texts.

4 Experiments

In order to test-drive the evaluation metric introduced in Section 3, several SMT systems showing effective, weak, poor or absent adaptation capability have been developed. Moreover, a preliminary investigation on backward reliability has been carried out. The next paragraphs detail and discuss the experiments performed.

4.1 Data

The task considered in this work involves the translation from English into Italian of documents in the Information Technology (IT) domain.

The training set consists of a large Translation Memory in the IT domain and several OPUS[6] subcorpora, namely KDE4, KDEdoc and PHP. The test set includes the human generated translation of 6 documents, disjoint from the training set. Although in the same domain, the test set is quite different from the training data as shown by comparing values of perplexity (650 vs. 40) and OOV rate (2.4% vs. 0.4%) computed on the source side.[7] Furthermore, the 6 documents significantly differ among each other: perplexity and OOV rate range from 465 to 880 and from 0.8 to 3.3, respectively. Table 1 collects overall statistics on training and test sets.

To simulate the stream of fresh data, the IT test set has been split into blocks of about a thousand[8] words each. Before splitting, sentences have been scrambled, with the rationale of generating a large number of homogeneous blocks, simulating a test set consisting of a single document.

[6]`http://opus.lingfil.uu.se`

[7]Figures for the training data were measured through a cross-validation technique.

[8]Different sizes have been also considered (three and five thousands) to test different adaptation rates, but results were qualitatively similar to those on shorter blocks and then are not reported.

4.2 Baseline System

The SMT baseline system is built upon the open-source MT toolkit Moses[9] (Koehn et al., 2007). The translation and the lexicalized reordering models are estimated on parallel training data with the default setting; a 5-gram LM smoothed through the improved Kneser-Ney technique (Chen and Goodman, 1999) is estimated on monolingual texts via the IRSTLM toolkit (Federico et al., 2008). Hereinafter, these models are referred to as background (BG) models. The log-linear interpolation weights are optimized by means of the standard MERT procedure provided within the Moses toolkit.

4.3 Adaptive System

The adapting SMT system is built on Moses as well. Besides the BG models of the baseline system, translation, reordering and language models estimated on the stream of fresh data are employed as additional features. Hereinafter, these models are referred to as foreground (FG) models. Unless differently specified, the FG models employed to translate a given block are trained on all preceding blocks. Note that the first instance of the adapting system (i.e. that translating the first block) is exactly the baseline system, because no adaptation data is available to train FG models yet. FG translation and reordering models are trained in the same way as the BG models. Due to the limited amount of adaptation data, the FG LM is a 3-gram LM smoothed through the more robust Witten-Bell technique (Witten and Bell, 1991).

The interpolation weights are inherited from a companion system trained and tuned on a different domain – official documents of the European Union organization – and are kept fixed.

4.4 Experiments on Adaptive SMT

First of all, the baseline and adapting systems were run on the scrambled test set and compared at both block-wise and incremental mode (see Section 2.3).

Figure 1 plots block-wise TER and BLEU scores of the baseline and adapting systems as functions of the amount (number of words) of adaptation data. On one hand, it can be guessed that the adapting system performs gradually better and better than the baseline; on the other hand, it is evident that such plots are not the most effective way to show the evolution of the adapting system. In fact, the translation difficulty of contiguous blocks can differ a lot. Hence, scores computed on them are not comparable and the corresponding curves are jagged.

The block-wise differences of TER and BLEU scores between the adapting and the baseline systems are plotted in Figure 2: the plots are now cleaner and more readable and vaguely suggest a positive trend, but still remain too jagged and do not provide any information about the absolute performance of the systems.

Figure 3 plots the incremental TER and BLEU scores of the baseline and adapting systems as functions of the amount of adaptation data. First of all, it is worth noting that the right-most values are the scores computed on the whole test set. In standard evaluation, those would be the only scores provided to show how the adapting system outperforms the baseline system; in particular, the relative improvement is larger for TER (9.3%) than for BLEU (3.9%) supposedly because tuning was performed to optimize BLEU score which thus is harder to improve. However, the overall scores obscure the way they are reached, that is the evolution over time of the systems, which is especially important for adaptive systems.

Secondly, the incremental evaluation yields much smoother plots clearly showing that after initial fluctuations: (i) performance of the baseline stabilizes around an average which does not change over time; (ii) scores of the adapting system tend to get increasingly better as more adaptation data is available for updating FG models.

The evaluation metric we are proposing, the percentage slope introduced in Section 3, is indeed able to spot such kind of paradigmatic behaviors as we will see in the next section. But before going on with the assessment of the metric, some further comments on Figure 3:

- in early stages, the adaptation is not effective, likely because of the scarcity of data. This raises two issues: design of more effective adaptation strategies and, in the CAT framework, identifying the appropriate time to replace the baseline with the adapting system;
- the adaptive system outperforms the baseline in

[9] http://www.statmt.org/moses

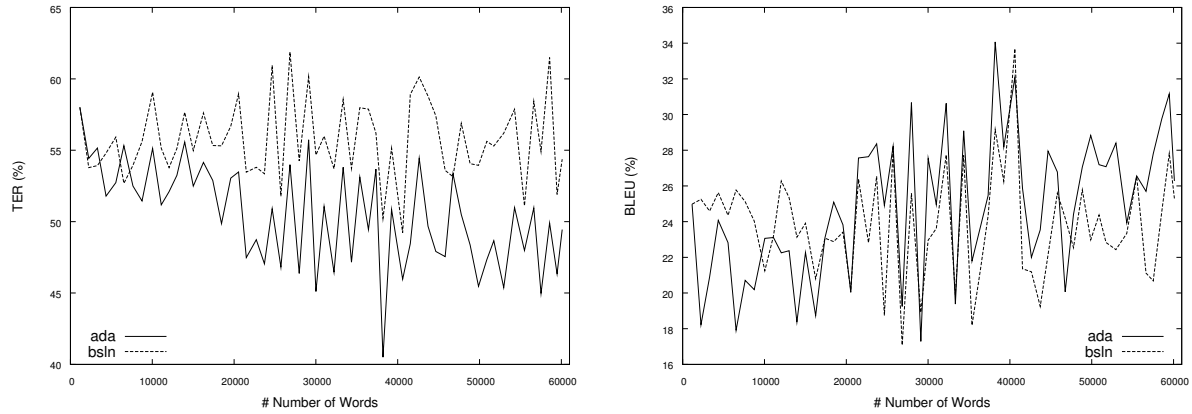

Figure 1: Block-wise TER (on the left) and BLEU (right) scores of the baseline and the dynamically adapting systems.

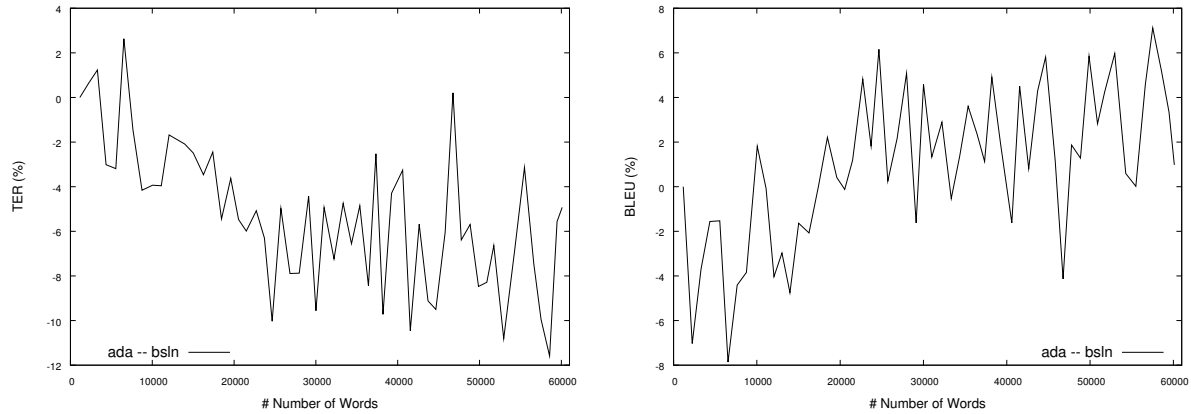

Figure 2: Block-wise TER (left) and BLEU (right) differences between the baseline and the dynamically adapting systems.

terms of TER very soon, while the overtaking with regard to BLEU is observed much later. This is because the baseline SMT system was tuned with respect to the BLEU score on in-domain data, differently to the adapting system.

Both these issues are out of the scope of this paper and will be subject of future investigations.

4.5 Assessment of the Percentage Slope

To assess its effectiveness, the percentage slope has been computed on errors committed by the baseline system, the adapting system and an adapting system featuring only FG models (that is without BG models). The FG-only system was used to translate each block either fairly and unfairly: the former mode fits the adaptation process sketched in Section 2.1; in the latter mode, the FG model is adapted on the block *before* its translation starts.

Figure 4 shows the TER and BLEU scores of such systems in the incremental evaluation. The four different behaviors are expected to correspond to different percentage slopes. In fact, the S values collected in Table 2 confirm the expectations:

- the baseline, completely unable to learn, has in fact an S of 100%
- the adapting system, that learns through a dynamic adaptation of FG models and generalizes thanks to BG models, has an S of 96-98%
- the FG-only adapting system tested in unfair mode worsens its performance as the models become larger, i.e. less focused on the block to be translated: this is evidenced by an S greater than 100%

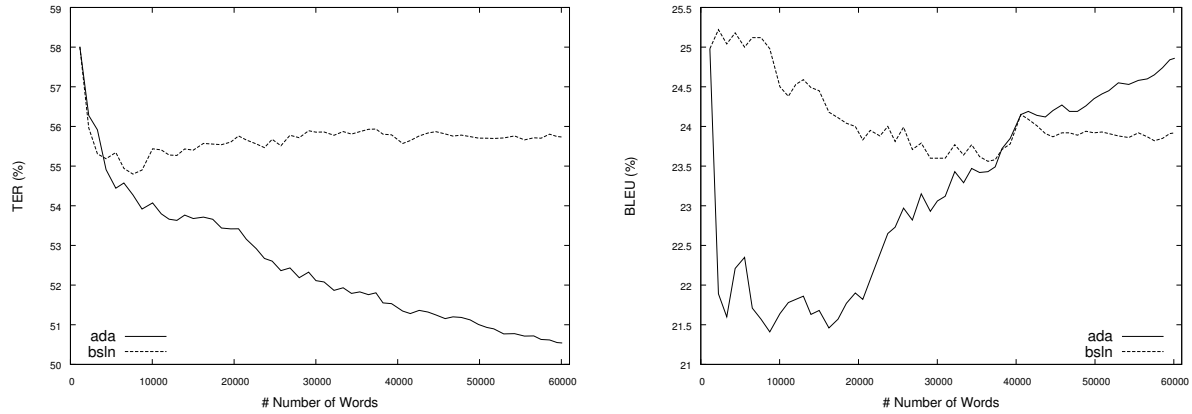

Figure 3: Incremental TER (left) and BLEU (right) scores of the baseline and the dynamically adapting systems.

model	system			
	baseline	adapting	FG-only adapting	
			fair	unfair
U	100.4	96.9	96.2	107.2
CA	100.3	97.7	96.5	107.4

Table 2: S values of 4 SMT systems (see text) for the block-wise TER evaluation, corresponding to the U model, and the incremental evaluation, corresponding to the CA model.

- the FG-only adapting system tested in fair mode increases its performance as the models become larger, i.e. more general, as evidenced by an S similar to that of our original adapting system (96%).

Therefore, we can state that S exposes common behaviors of evolving SMT systems; however, standard metrics like TER and BLEU are still in charge of providing absolute performance measures.

In order to give a hint for properly interpreting the values reported, we summarize the discussion in (Stump P.E., 2002) about "typical learning slopes". Operations that are fully automated tend to have slopes of 100%, 70% if entirely manual, an intermediate value if mixed. In real industrial environments, the average slope depends on the type of manufacturing activity: for example, in aircraft industry it is about 85%, it ranges in 90-95% in electronics and in machining. Hence, a 96-98% slope as we measured in our experiments must be considered a significant learning ability of a fully automated system.

4.6 Experiments on Backward Reliability

A proper assessment of the backward reliability of an evolving system as defined in Section 2.2 would require the identification of patterns translated differently by the system during its life. We will investigate this issue in the future. For the moment, we try to attack the problem from a global point of view: we simply check that the adaptive system does "remember" its previous translation capabilities "on average", while it learns to better translate novel texts.

To this end, a cross-validation policy was followed: the first two thirds of each test set document are used for dynamically training the FG models, while the remaining portions are used as held-out test sets.

Figure 5 reports the TER and BLEU scores on the 6 test sets of three systems: the baseline system (`bsln`), the adapting system (`ada`) fed by incrementally merging the available reduced adaptation sets, and the system adapted on all adaptation data sets (`final`).

The `final` system achieves performance close to `ada` system on each held-out set; this reveals that our adaptation process is effective both in learning and in remembering.

We think that the monitoring of the backward reliability of adapting systems is a good practice. A cross validation scheme like ours allows not only to reveal the backward reliability as shown before, but also to discover the forgetting trend of, for example, an MT system featuring an overly aggressive learn-

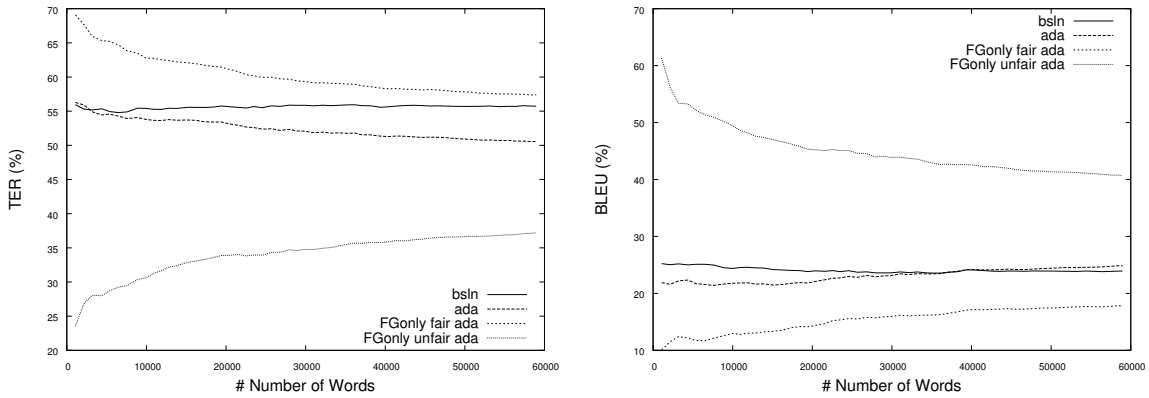

Figure 4: Incremental TER (left) and BLEU (right) of 4 systems showing different learning slopes.

ing method. On the other hand, it only provides cues about the average behavior and it is not as quickly informative as a single score could be. Hence, the design of a proper metric for measuring the backward reliability of MT systems is a challenging task that should be faced by the research community.

5 Summary and Future Work

The evaluation of a dynamically adapting system is an open issue. Metrics used in interactive MT such as HTER or field tests, are infeasible in the daily development as they involve human translators/judges. On the other hand, standard MT evaluation metrics either do not expose changes over time (BLEU, TER) or cannot be applied (CER).

The main contribution of this paper is to propose the use of the percentage slope for the evaluation of adapting MT systems, a metric borrowed from the theory on learning curves. For assessing its effectiveness, we have developed a simple but effective adapting SMT system suitable to work in the context of a CAT tool supported by MT. We have compared several ways to plot the change in error rate over time for different systems and identified the most suitable for computing the percentage slope. Finally, we have shown that the percentage slope well exposes the paradigmatic behaviors of evolving SMT systems.

The MateCAT project has scheduled field tests for the near future which will allow for inclusion of human productivity in the assessment of the percentage slope. Moreover, efforts will be devoted to the design of adaptation techniques which are more sophisticated than the simple approach used in this work.

We have also identified the issue of backward reliability of an adapting system, that is the ability to learn without forgetting the past, and the importance of monitoring it. A best practice based on a cross validation scheme has been proposed. Future investigations will concern finding an effective metric to measure backward reliability.

Acknowledgments

This work was supported by the MateCAT project, which is funded by the EC under the 7^{th} Framework Programme.

References

N. Cesa-Bianchi, G. Reverberi, and S. Szedmak. 2008. Online learning algorithms for computer-assisted translation. Deliverable 4.2, SMART project (FP6). `http://www.smart-project.eu/files/D42.pdf`.

S. F. Chen and J. Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 4(13):359–393.

M. Federico, N. Bertoldi, and M. Cettolo. 2008. IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models. In *Proc. of Interspeech*, pp. 1618–1621, Melbourne, Australia.

G. Foster and R. Kuhn. 2007. Mixture-Model Adaptation for SMT. In *Proc. of WMT*, pp. 128–135, Prague, Czech Republic.

S. Khadivi. 2008. *Statistical Computer-Assisted Translation*. Ph.D. thesis, RWTH Aachen University,

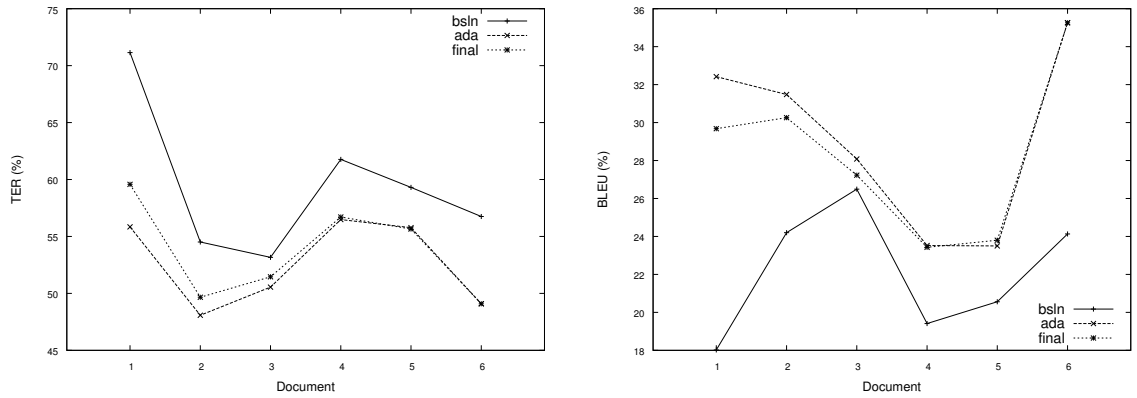

Figure 5: TER (left) and BLEU (right) scores of the baseline system, the evolving system and the final adapted system on the document-specific held-out test sets.

Aachen, Germany. Advisors: Hermann Ney and Enrique Vidal.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proc. of ACL: Demo and Poster Sessions*, pp. 177–180, Prague, Czech Republic.

E. Macklovitch. 2006. Transtype2: The last word. In *Proc. of LREC 2006*, Genoa, Italy.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Research Report RC22176, IBM Research Division, Thomas J. Watson Research Center.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA*, Boston, US-MA.

E. Stump P.E. 2002. All about learning curves. In *Proc. of SCEA*. `http://www.galorath.com/images/uploads/LearningCurves1.pdf`.

I. H. Witten and T. C. Bell. 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Trans. Inform. Theory*, IT-37(4):1085–1094.