Generative and Discriminative Methods for Online Adaptation in SMT

K. Wäschle † P. Simianer † N. Bertoldi ‡ S. Riezler † M. Federico‡

† Department of Computational Linguistics
Heidelberg University
69120 Heidelberg, Germany
{surname}@cl.uni-heidelberg.de

‡ Fondazione Bruno Kessler
via Sommarive 18
38123 Trento, Italy
{surname}@fbk.eu

Abstract

In an online learning protocol, immediate feedback about each example is used to refine the next prediction. We apply this protocol to statistical machine translation for computer-assisted translation and compare generative and discriminative approaches for online adaptation. We develop our methods on reference translations and test on feedback gathered from professional translators. Experimental results show that improvements of straightforward adaptations of translation and language model are greater than those achieved by discriminative re-ranking. However, the improvements add up to 4 BLEU points over a baseline static model.

1 Introduction

State-of-the-art Statistical Machine Translation (SMT) systems translate each sentence of an input text in isolation. While this reduces the complexity of translating large documents, it introduces the problem that information beyond the sentence level is lost. In this paper, we investigate a scenario where after producing a system translation, user feedback in form of a manual translation or in form of a user correction is available to refine the prediction. This scenario fits well into a online learning protocol (Cesa-Bianchi and Lugosi, 2006), where a stream of input data is revealed to the learner one by one. For each input example, the learner must make a prediction, after which the actual output is revealed, which the learner can use to refine the next prediction. We apply this protocol to a computer-assisted translation (CAT) scenario. While SMT is not yet able to provide output that is suitable for publication without human intervention, CAT is an area where the missing human feedback is available. From the viewpoint of professional translators, immediate refinement of the SMT system in response to user feedback is crucial in order to offer the user experience of a system that learns from feedback and corrections. Online adaptation achieves this by increasing consistency of system translations with respect to the user translation of previously seen examples.

We present and compare two approaches to online adaptation that can be applied in a CAT scenario. We propose methods that augment the generative components of the SMT system, translation and language model, straightforwardly by building local models of phrases and n-grams from user feedback. Furthermore, we present a discriminative method based on a structured perceptron to refine a feature-based re-ranking module applied to the k-best translations of the SMT system.

To our knowledge, this is the first comparison of generative and discriminative online adaptation methods in a CAT scenario. The discriminative approach allows to perform feature development and training independently of the underlying SMT system. In the generative approach, the model is simple, however, updates have to be communicated to the decoder. In sum, the gains of both approaches add up to average BLEU improvements of 4 points over a baseline non-adapted model.

2 Previous Work

Online learning methods in SMT are found in the context of *stochastic methods for discriminative training* (Liang et al., 2006; Chiang et al., 2008),

Sima'an, K., Forcada, M.L., Grasmick, D., Depraetere, H., Way, A. (eds.)
Proceedings of the XIV Machine Translation Summit (Nice, September 2–6, 2013), p. 11–18.

or *streaming* scenarios for incremental adaptation of the core components of SMT (Levenberg et al., 2010). However, the online learning protocol is applied in these approaches to training data only, i.e., parameters are updated on a per-example basis on the training set, while testing is done by re-translating the full test set using the final model.

Further related work can be found in the application of incremental learning to *domain adaptation in SMT*. Here a *local* and a *global* model have to be combined, either in a log-linear combination (Koehn and Schroeder, 2007), with a fill-up method (Bisazza et al., 2011), or via ultraconservative updating (Liu et al., 2012).

Carpuat and Simard (2012) show that increased *translation consistency* does not correlate with better translation quality, however, translation errors are indicated by inconsistencies. Our approach can be seen as a successful approach to improve translation quality by enforcing local consistency through online learning.

Cesa-Bianchi et al. (2008) are the first to apply *online discriminative re-ranking* to a CAT scenario. Incremental adaptations of the *generative components* of SMT have been presented for a related scenario, *interactive machine translation*, where an MT component produces hypotheses based on partial translations of a sentence (Nepveu et al., 2004; Ortiz-Martínez et al., 2010). Our online learning protocol is similar, but operating on the sentence instead of word or phrase level.

Incremental adaptations have also been presented for larger batches of data (Bertoldi et al., 2012). In terms of granularity, our scenario is most similar to the work by Hardt and Elming (2010), where the Moses training procedure is employed to update the phrase table immediately after a reference becomes available. Our work, however, focuses on adapting both language and translation model with techniques where the global model remains unchanged. This is important in a CAT scenario, where several users might use the same global model but individual local models.

3 Online Adaptation in SMT

Cesa-Bianchi and Lugosi (2006) presented a protocol for online learning with expert advice. This protocol can be adapted to our scenario of online adaptation in SMT as follows:

```
Train global model $M_g$
for each document $d$ of $|d|$ sentences
    Reset local model $M_d = \emptyset$
    for each example $t = 1, \ldots, |d|$
        0. Combine $M_g$ and $M_d$ into $M_{g+d}$
        1. Receive input sentence $x_t$
        2. Output translation $\hat{y}_t$ from $M_{g+d}$
        3. Receive user translation $y_t$
        4. Refine $M_d$ on pair $(x_t, y_t)$
```

The learning process starts from training a global model M_g on parallel data in the range of millions of sentence pairs. Then for each document d, consisting of a few hundred up to a thousand sentences, a local model M_d is created. For each example, first the static global model M_g and the current local model M_d are combined into a model M_{g+d}. Then the input x_t is translated into $\hat{y}_t$ using the model M_{g+d}. Finally the local model M_d is refined on feedback y_t that is received immediately after producing $\hat{y}_t$.

Evaluations reported in this paper take the local predictions $\hat{y}_t$ and compare them to the user translations y_t for each document, e.g., using $BLEU\{(\hat{y}_t, y_t)\}_{t=1}^{|d|}$ (Papineni et al., 2002). Note that this setup differs from the more standard scenario where the whole test set is re-translated using the learned model. However, the evaluation in our online learning scenario is still fair since only feedback from previous test set examples is used to update the current model.

We present three techniques for refinements of local SMT models (step 4), namely adaptations of the generative components of translation model (TM) (Section 3.3) and language model (LM) (Section 3.4) and adaptation via discriminative re-ranking (Section 3.5). Different refinements result in different modes of combination of *global* and *local* models (step 0). Both generative and discriminative adaptation modes deploy a constrained search technique (Section 3.2) to extract information relevant for system refinement from the received user feedback (step 3). Translation (step 2) employs a standard phrase-based SMT engine.

3.1 Baseline System

The MT engine is built upon the open source toolkit Moses (Koehn et al., 2007). The global translation and the lexicalized reordering models are estimated on parallel training data with default

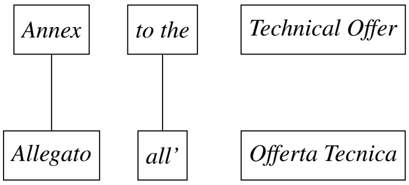

Figure 1: Phrase segmentation and alignment.

setting. The global 5-gram LM smoothed through the improved Kneser-Ney technique is estimated on the target monolingual side of the parallel training data using the IRSTLM toolkit (Federico et al., 2008). Models are case-sensitive. The log-linear interpolation weights are optimized using the standard MERT procedure provided with the Moses toolkit. The baseline system also provides a list of k-best translations. In the online discriminative re-ranking approach, this k-best list is rescored according to lexicalized sparse features including phrase pairs and target-side n-grams.

3.2 Constrained Search for Feedback Exploitation

In order to extract information for system refinement from user feedback, source and user translation need to be aligned at the phrase-level. We use a constrained search technique described in Cettolo et al. (2010) to achieve this, which optimizes the coverage of both source and target sentences given a set of translation options.

The search produces exactly one phrase segmentation and alignment, and allows gaps such that some source and target words may be uncovered. Unambiguous gaps (i.e. one on the source and one on the target side) can then be aligned. It differs in this respect from forced decoding which produces an alignment only when the target is fully reachable with the given models.

From the phrase alignment, three types of phrase pairs can be collected: (i) **new** phrase pairs by aligning unambiguous gaps; (ii) **known** phrase pairs already present in the given model; (iii) **full** phrase pairs consisting of the complete source sentence and its user translation. Only phrases that contain at least one content word are considered.

From the alignment shown in Figure 1, we extract the new phrase pair *Technical Offer* → *Offerta Tecnica*, the known phrase pairs *Annex* → *Allegato* and *to the* → *all'* and the full phrase *Annex to the Technical Offer* → *Allegato all' Offerta Tecnica*.

3.3 TM Adaptation

The growing collection of source sentences and corresponding user-approved translations enables the construction of a *local* translation model. The goal of this local model is to reward MT translations that are consistent with previous user translations as well as to integrate new translations learned from user corrections, in order to better translate the following sentences. From each sentence pair, all phrase pairs extracted with the constrained search technique described in Section 3.2 are inserted into a cache; probabilities are estimated based on the relative frequency of the target phrase given the source phrase within the cache. Cache and model are updated on a per-sentence basis as soon as source sentence and user translation become available.

A fast way to integrate the constantly changing local model in the decoder at run-time is the Moses XML input option. Translation options for phrases can be passed to the decoder in XML-like markup. Multiple phrase translations and their corresponding probabilities for a source phrase can be suggested:

Annex to the `<p translation="`*Offerta Tecnica*`||`*Proposta Tecnica*`" prob="0.75||0.25">`*Technical Offer*`</p>`

Moses offers two ways to interact with this local phrase table. In *inclusive* mode, the given phrase translations compete with existing phrase table entries. The decoder is forced to choose only from the given translations in *exclusive* mode. During development, we found that the exclusive option is too strict in our scenario. Though most phrase pairs are correct and useful additions, for example spelling variants such as *S.p.A*→*SpA* or domain vocabulary such as *lease payment*→*canone*, some are restricted to a specific context, e.g. translation from singular to plural such as *service*→*servizi*, and some are actually incorrect. In inclusive mode, the global translation and language model can reject unlikely translations.

Since the XML input option does not support overlapping phrases, sentences are annotated in a greedy way from left to right. Only phrases that contain at least one content word are considered. For each phrase in the input sentence, the cache is checked for possible translations, starting from the complete sentence down to single words. In this way, translations for larger spans are preferred over word translations. We did not explore other setups, such as preferring newly learned phrases over older options from the cache, but instead opted to keep the implementation simple.

3.4 LM Adaptation

Similar to the local translation model, we build a local language model to reward target n-grams seen in user translations. This is implemented by an additional feature in the log-linear model, which computes an additional score for each translation option based on a target n-gram cache. n-grams are associated with an age and a score, which is strictly positive and decays exponentially as the age increases. The cache is filled dynamically using XML-like input as follows:

<dlt cblm=*"Offerta Tecnica*||*Offerta*|| *Tecnica"*/>*Annex to the Technical Offer*

At each iteration, we update the cache with all user translation target n-grams that contain at least one content word. n-grams are added with age of 1 and the age of the existing entries is increased by 1. The local LM score is computed as follows: Given a translation option, the scores of all substrings included in the cache are summed up. Any string not found in the cache receives a score of 0, i.e. no reward. n-grams crossing over contiguous translation options are not taken into account. Note that the proposed feature is simply a stateless function which rewards approved translation options, which are expected to be of high quality.

To control the influence of the local language model, the additional weight is optimized with the Simplex algorithm; weights of the baseline system tuned with MERT are taken as fixed.

3.5 Online Discriminative Re-Ranking

The learner used in our online discriminative re-ranking approach is a structured perceptron (Collins, 2002). We use lexicalized sparse features defined by two feature templates: First, all phrase pairs found by the decoder (for system translations) or by the constrained search (for the user translation) are used as features. Second, we use features defined by target-side n-grams from $n = 1, \ldots, 4$ in the user translation. Our features are not indicator functions, but use the number of source words (for the first type of features) and the number of words in target-side n-grams (for the second type of features) as values. Given a feature representation $f(x, y)$ for a source-target pair (x, y), and a corresponding weight vector w, the perceptron update on a training example (x_t, y_t) where the prediction $\hat{y} = \arg\max_y \langle w, f(x_t, y)\rangle$ does not match the target y_t is defined as:

$$w = w + f(x_t, y_t) - f(x_t, \hat{y})$$

The constrained search allows us to perform updates even on translations that are not reachable by the decoder. For the purpose of discriminative training, in our setup all references are reachable since we can extract features from them and assign them model scores.

4 Experimental Evaluation

We develop and test our system on English-Italian data from the IT domain. In addition, we show that the results transfer well to other domains and language pairs by evaluating a system trained on German-English patent text on patent documents.

The training data for the global IT models is compiled from a translation memory and several OPUS[1] corpora related to the Information Technology domain. For development, six documents corresponding to IT projects are used, where the reference is considered to be a user-approved translation. We split the documents into two groups, dev1 and dev2. Weight optimization was performed on dev1; the best overall system configuration was determined according to the scores computed on dev2.

For testing, one document is taken into account for the IT domain, for which actual user corrections from three different translators (A-C) are available for each sentence that were collected during a field test. We report the scores for all three translators, regarding each translator as a document. This choice has strong motivations in the online adaptation scenario. Each translator processed the sentences in his or her preferred order, and provided a different reference. Consequently, the original baseline system evolves differently, and possibly achieves different performance. We aim to show that the proposed techniques give consistent improvement among different sets of user

[1] http://opus.lingfil.uu.se

	IT		patent	
	doc	sentences	doc	sentences
	train	1,167 K	train	4,199 K
dev1	prj1	420	pat1	300
	prj2	931	pat2	227
	prj3	375	pat3	239
dev2	prj4	289		
	prj5	1,183		
	prj6	864		
test	prj7A	176	pat4	232
	prj7B	176	pat5	230
	prj7C	176	pat6	225
			pat7	231

Table 1: Statistics for training, dev and test data.

feedback, regardless the overall performance of the baseline system.

As evidence that the results transfer to different languages and domains, we train a system on German-English patent text sampled from title, abstract and description sections from the PatTR[2] corpus (Wäschle and Riezler, 2012). We tune the weight for the cache-based language model feature on a domain-specific set dev1, consisting of three patents containing title, abstract and description sections, but take the best overall system configuration determined on the IT dev2 set. We report evaluation results on four more patent documents. Statistics for all data are reported in Table 1.

4.1 Evaluation Setup

The proposed approaches were evaluated with BLEU (Papineni et al., 2002), exploiting the user feedback as reference. We report mean BLEU scores on IT dev1, IT dev2 and patent test set. In all tables, best results are highlighted in **bold** face. We report mean improvement over the baseline in small font size and give the standard deviation of the improvements over all documents in the group in square brackets. Statistical significance is assessed using approximate randomization (Noreen, 1989).

Our evaluation matches local predictions against the test set references, i.e. testing is not done by re-translating the test set with the final model. Instead, feedback from previous examples is used to update the model before the translation of the current example, which is then used for testing.

	tm	1gr lm	4gr lm	tgt lm
IT dev1	25.49	27.32	**27.64**	26.36
		+1.83 [0.45]	**+2.15** [0.76]	+0.87 [0.74]
IT dev2	23.91	24.62	**25.25**	24.87
		+0.70 [1.17]	**+1.34** [1.26]	+0.95 [1.01]

Table 3: LM adaptation using 1gr lm, 4gr lm, and tm-tgt lm conditions (see Section 3.4) on top of best TM adaptation (tm). Figures reported are mean BLEU scores and mean differences and standard deviation from baseline (in small font size).

The previous metrics provide absolute performance of the system, but they are not actually sufficient in an online adaptation scenario, in which the system evolves dynamically over time. As shown in Bertoldi et al. (2012), adapting systems can be effectively analyzed by means of the *percentage slope*, which measures their learning capability. In our investigation similar learning capability (and percentage slope) was observed if different automatic MT metrics are taken into account. We report the percentage slope S computed on the cumulative BLEU score only, starting at 30 sentences to get reliable results.

4.2 Local Translation Model

Table 2 shows BLEU scores and differences from the baseline on IT dev1 and IT dev2 sets for TM adaptation as described in Section 3.3. We see that each condition of TM adaptation, new, known or full, yields individual improvements. Furthermore, improvements of individual conditions add up to an overall improvement of 2.90 (IT dev1) and 2.42 (IT dev2) BLEU points over the baseline for the combination of all conditions, namely new+known+full.

4.3 Local Language Model

Table 3 shows a comparison of different LM adaptation conditions on top of the best TM adaptation (new+known+full) given in Table 2. Using n-grams up to order 4 (4gr lm) yields an additional 2.15 (IT dev1) and 1.34 (IT dev2) BLEU points. Using only 1-gram features (1gr lm) or rewarding only those n-grams that are target sides of phrase pairs (tm-tgt lm) only give half the improvement.

[2] http://www.cl.uni-heidelberg.de/statnlpgroup/pattr/

	bsln	new	known	full	new+known	new+known+full
IT dev1	22.59	23.11	23.73	24.22	24.33	**25.49**
		+0.52 [0.57]	+1.14 [0.70]	+1.63 [1.73]	+1.75 [0.80]	**+2.90** [2.18]
IT dev2	21.49	21.64	22.24	23.07	22.42	**23.91**
		+0.15 [0.06]	+0.75 [0.15]	+1.58 [0.91]	+0.93 [0.19]	**+2.42** [0.83]

Table 2: TM adaptation using new, known or full phrases (see Section 3.3) and combinations of the three on IT dev1 and IT dev2 on top of the baseline. Figures reported are mean BLEU scores and mean difference and standard deviation from baseline (in small font size).

	bsln	known+lm	rerank
IT dev1	22.59	**25.78**	23.74
		+2.69 [1.68]	+1.15 [0.82]
IT dev2	21.49	**23.43**	22.85
		+1.94 [1.41]	+1.36 [0.65]

Table 4: Adaptation of generative models (known+lm) vs. discriminative re-ranking (rerank) on IT dev1 and IT dev2. Figures reported are mean BLEU scores and mean differences and standard deviation from baseline (in small font size).

	bsln		tm+lm	
	BLEU	S	BLEU	S
prj7A	41.10	94.71	**42.97 +1.87**	93.60 -1.11
prj7B	39.68	98.56	39.72 +0.04	97.63 -0.93
prj7C	30.68	99.87	**33.76 +3.08**	97.62 -2.25

Table 5: Main results for TM and LM adaptation on IT test set featuring user corrections by three translators each (A-C). Figures reported are BLEU scores and percentage Slope S with differences from baseline (in small font size).

4.4 Discriminative Re-Ranking

To compare the adaptation of generative models with our discriminative re-ranking approach, we conducted an experiment with similar settings for both approaches: TM adaptation was limited to the known phrases setup, i.e., weights of phrases in the global model are re-weights if they are found in local updating; LM adaptation was set to use n-grams (n up to 4). Discriminative re-ranking used phrase-pair and n-gram features (n up to 4) as described in Section 3.5. The size of the k-best list of translations was set to $k = 100$. The gain of online discriminative re-ranking, reported in Table 4, is significant: +1.15 (IT dev1) and +1.36 (IT dev2) BLEU points over the baseline. However, adaptation of TM and LM shows larger gains, despite using similar information. We conjecture that a direct interaction with the decoder is beneficial over offline re-ranking even if similar information is used.

4.5 Main Results

The main results of our online adaptation experiments are shown in Tables 5 and 6. On the IT test set we find an average[3] improvement of about 2 BLEU points for the combination of TM and LM adaptation over a static baseline model. This corresponds to the findings on the IT dev sets. Similar improvements are gained on the patent test set for discriminative reranking (+2.28 BLEU) and combined TM and LM adaptation (+2.98 BLEU). Furthermore, we see that the improvements of online adaptation for discriminative and generative models are additive, yielding a cumulative improvement of +3.76 BLEU points. All improvements over the baseline (except for translator B in Table 5) are statistically significant.

One goal of the proposed online adaptation approaches is the improvement of the system over time. Therefore, the considered systems are also evaluated in terms of percentage slope (S), which measures their learning capability as explained in Section 4.1. A system improves its performance over time as much as S is lower than 100, under the assumption that the difficulty of the test set is homogeneous. According to the figures reported in Table 5, the baseline system shows an improv-

[3] Even though the source document is the same in all cases, baseline and system results vary depending on the user, since each translator produced different reference translations.

	bsln	rerank	tm+lm	tm+lm +rerank
pat4-7	30.26	32.54	33.24	**34.02**
		+2.28 [1.47]	+2.98 [2.03]	**+3.76 [2.08]**

Table 6: Main results for generative and discriminative adaptation and combination of both on patents. Figures reported are mean BLEU scores over four test documents and mean difference and standard deviation from baseline (in small font size).

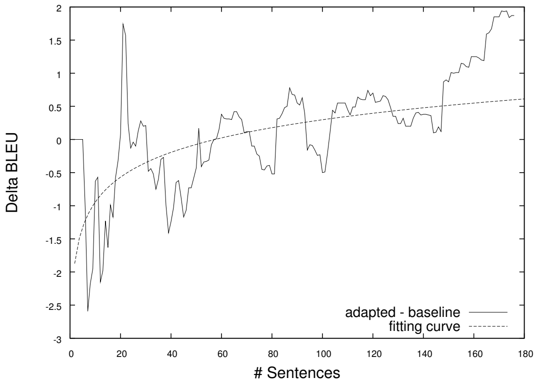

Figure 2: Cumulative BLEU difference between best-adapted and baseline system on prj7A.

ing trend, which is unexpected, because it does not adapt over time. This reveals that the document actually diminishes its complexity sentence after sentence. Therefore, the percentage slopes of the baseline can be considered as an offset against other systems can be compared. Taking into account the baseline as offset, both tm and tm+lm system have consistent learning capability.

Figure 2 plots the difference between the cumulative BLEU scores of the tm+lm and baseline systems on prj7A; the plot gives a graphical evidence that the online adaptation approach is effective. Although on some parts of the document the adapted system slightly worsens, e.g. for sentences 70–80 and 90-100, the general improving trend is documented by the fitting learning curve.

5 Discussion

Figure 3 shows a comparison of translation output for a static baseline and generative adaptation. The adapted system is able to correctly predict the user translation.

To analyze the performance of the discriminative approach, we examined positively and negatively weighted features that explain how reranking can help the system recover from errors by reweighting translations. For example, our reranking model captures the contextual difference of translating the English *and* into the Italian *e* before a consonant or *ed* before a vowel by assigning high positive weight to n-grams such as *DLI ed IBM* and *ed IBM* and a high negative weight to n-grams such as *DLI e IBM* and *e IBM*. Due to the frequent use of title case in the IT data, the system also learned to prefer phrase pairs with matching case (*Life* → *Vita*, *machine* → *macchina*) over pairs with case mismatch (*Customer* → *clienti*).

6 Conclusion

We presented an application of an online learning protocol to SMT. The protocol offers immediate feedback after each translation output, and it allows an SMT system to learn from this feedback for future translations. Assuming coherent texts, the obvious advantage of this scenario is the possibility to improve consistency of translations by learning from successive feedback and corrections. While this setup might be restricted, it naturally fits a CAT scenario where feedback is provided by professional translators. Immediate refinement of the SMT system upon supplying feedback is crucial in order to offer a user experience of working with a system that learns from feedback and corrections.

We compared two approaches to online adaptation. The advantages of the discriminative approach lies in the offline computability of features from k-best translations, and in a training process that does not need to communicate with the decoder. The generative approach has to update the decoder with new information about phrases, n-grams, and weights, however, this overhead is minimal compared to the larger gains due to generative adaptation. If translation quality is the main concern, stacking of discriminative and generative online adaptation leads to additive improvements. On patent data we found gains of 4 BLEU points by combined generative and discriminative online adaptation over a static baseline.

In future work, we plan to investigate advanced methods for online adaptation and enhanced approaches to extract information from the user feedback, such as new phrase alignment methods. Furthermore, we intend to run interactive field tests in a real-world CAT setting, in order to conduct a

source	*A copy type is automatically assigned to a consistency group.*
baseline	*Una copia tipo viene automaticamente assegnato a un gruppo di coerenza.*
adapt **tm**	*Una copia tipo viene automaticamente assegnato a un gruppo di* ***congruenza****.*
adapt **tm+lm**	*Un* ***tipo di copia*** *viene automaticamente assegnato a un gruppo di* ***congruenza****.*
reference	*Un tipo di copia viene assegnato automaticamente a un gruppo di congruenza.*

Figure 3: Comparison of baseline and adapted system output on test set. Using an adapted translation model, the system is able to correct the translation of *consistency* . By adding language model adaptation, the system also produces a correct Italian compound word for the translation of *copy type*.

timing comparison between different online adaptation techniques and to test the robustness of our methods and their reliability over time.

Acknowledgments

FBK researchers were supported by the MateCAT project, funded by the EC under FP7; researchers at Heidelberg University by DFG grant “Cross-language Learning-to-Rank for Patent Retrieval”.

References

Bertoldi, Nicola, Mauro Cettolo, Marcello Federico, and Christian Buck. 2012. Evaluating the learning curve of domain adaptive statistical machine translation systems. In *WMT'12*.

Bisazza, Arianna, Nick Ruiz, and Marcello Federico. 2011. Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation. In *IWSLT'11*.

Carpuat, Marine and Michel Simard. 2012. The trouble with SMT consistency. In *WMT'12*.

Cesa-Bianchi, Nicolò and Gàbor Lugosi. 2006. *Prediction, Learning, and Games*. Cambridge University Press.

Cesa-Bianchi, Nicolò, Gabriele Reverberi, and Sandor Szedmak. 2008. Online learning algorithms for computer-assisted translation. Technical report, SMART (`www.smart-project.eu`).

Cettolo, Mauro, Marcello Federico, and Nicola Bertoldi. 2010. Mining parallel fragments from comparable texts. In *IWSLT'10*.

Chiang, David, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *EMNLP'08*.

Collins, Michael. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *EMNLP'02*.

Federico, Marcello, Nicola Bertoldi, and Mauro Cettolo. 2008. IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models. In *Interspeech'08*.

Hardt, Daniel and Jakob Elming. 2010. Incremental re-training for post-editing SMT. In *AMTA'10*.

Koehn, Philipp and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *WMT'07*.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Birch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL'07*.

Levenberg, Abby, Chris Callison-Burch, and Miles Osborne. 2010. Stream-based translation models for statistical machine translation. In *HLT-NAACL'10*.

Liang, Percy, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *COLING-ACL'06*.

Liu, Lemao, Hailong Cao, Taro Watanabe, Tiejun Zhao, Mo Yu, and CongHui Zhu. 2012. Locally training the log-linear model for SMT. In *EMNLP'12*.

Nepveu, Laurent, Guy Lapalme, Philippe Langlais, and George Foster. 2004. Adaptive language and translation models for interactive machine translation. In *EMNLP'04*.

Noreen, Eric W. 1989. *Computer Intensive Methods for Testing Hypotheses. An Introduction*. Wiley.

Ortiz-Martínez, Daniel, Ismal García-Varea, and Francisco Casacuberta. 2010. Online lerning for interactive statistical machine translation. In *HLT-NAACL'10*.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL'02)*.

Wäschle, Katharina and Stefan Riezler. 2012. Analyzing Parallelism and Domain Similarities in the MAREC Patent Corpus. In *IRFC'12*.