A Coactive Learning View of Online Structured Prediction in Statistical Machine Translation

Artem Sokolov and **Stefan Riezler**[*]
Computational Linguistics & IWR[*]
69120 Heidelberg, Germany
{sokolov,riezler}@cl.uni-heidelberg.de

Shay B. Cohen
University of Edinburgh
Edinburgh EH8 9LE, UK
scohen@inf.ed.ac.uk

Abstract

We present a theoretical analysis of online parameter tuning in statistical machine translation (SMT) from a coactive learning view. This perspective allows us to give regret and generalization bounds for latent perceptron algorithms that are common in SMT, but fall outside of the standard convex optimization scenario. Coactive learning also introduces the concept of weak feedback, which we apply in a proof-of-concept experiment to SMT, showing that learning from feedback that consists of slight improvements over predictions leads to convergence in regret and translation error rate. This suggests that coactive learning might be a viable framework for interactive machine translation. Furthermore, we find that surrogate translations replacing references that are unreachable in the decoder search space can be interpreted as weak feedback and lead to convergence in learning, if they admit an underlying linear model.

1 Introduction

Online learning has become the tool of choice for large scale machine learning scenarios. Compared to batch learning, its advantages include memory efficiency, due to parameter updates being performed on the basis of single examples, and runtime efficiency, where a constant number of passes over the training sample is sufficient for convergence (Bottou and Bousquet, 2004). Statistical Machine Translation (SMT) has embraced the potential of online learning, both to handle millions of features and/or millions of data in parameter tuning via online structured prediction (see Liang et al. (2006) for seminal early work), and in interactive learning from user post-edits (see Cesa-Bianchi et al. (2008) for pioneering work on online computer-assisted translation). Online learning algorithms can be given a theoretical analysis in the framework of online convex optimization (Shalev-Shwartz, 2012), however, the application of online learning techniques to SMT sacrifices convexity because of latent derivation variables, and because of surrogate translations replacing human references that are unreachable in the decoder search space. For example, the objective function actually optimized in Liang et al.'s (2006) application of Collins' (2002) structure perceptron has been analyzed by Gimpel and Smith (2012) as a non-convex ramp loss function (McAllester and Keshet, 2011; Do et al., 2008; Collobert et al., 2006). Since online convex optimization does not provide convergence guarantees for the algorithm of Liang et al. (2006), Gimpel and Smith (2012) recommend CCCP (Yuille and Rangarajan, 2003) instead for optimization, but fail to provide a theoretical analysis of Liang et al.'s (2006) actual algorithm under the new objective.

The goal of this paper is to present an alternative theoretical analysis of online learning algorithms for SMT from the viewpoint of coactive learning (Shivaswamy and Joachims, 2012). This framework allows us to make three main contributions:

- Firstly, the proof techniques of Shivaswamy and Joachims (2012) are a simple and elegant tool for a theoretical analysis of perceptron-style algorithms that date back to the perceptron mistake bound of Novikoff (1962). These techniques provide an alternative to an online gradient descent view of perceptron-style algorithms, and can easily be extended to obtain regret bounds for a la-

Proceedings of the 19th Conference on Computational Language Learning, pages 1–11,
Beijing, China, July 30-31, 2015. ©2015 Association for Computational Linguistics

tent perceptron algorithm at a rate of $O\left(\frac{1}{\sqrt{T}}\right)$, with possible improvements by using re-scaling. This bound can be directly used to derive generalization guarantees for online and online-to-batch conversions of the algorithm, based on well-known concentration inequalities. Our analysis covers the approach of Liang et al. (2006) and supersedes Sun et al. (2013)'s analysis of the latent perceptron by providing simpler proofs and by adding a generalization analysis. Furthermore, an online learning framework such as coactive learning covers problems such as changing n-best lists after each update that were explicitly excluded from the batch analysis of Gimpel and Smith (2012) and considered fixed in the analysis of Sun et al. (2013).

- Our second contribution is an extension of the online learning scenario in SMT to include a notion of "weak feedback" for the latent perceptron: Coactive learning follows an online learning protocol, where at each round t, the learner predicts a structured object y_t for an input x_t, and the user corrects the learner by responding with an improved, but not necessarily optimal, object $\bar{y}_t$ with respect to a utility function U. The key asset of coactive learning is the ability of the learner to converge to predictions that are close to optimal structures y_t^*, although the utility function is unknown to the learner, and only weak feedback in form of slightly improved structures $\bar{y}_t$ is seen in training. We present a proof-of-concept experiment in which translation feedback of varying grades is chosen from the n-best list of an "optimal" model that has access to full information. We show that weak feedback structures correspond to improvements in TER (Snover et al., 2006) over predicted structures, and that learning from weak feedback minimizes regret and TER.

- Our third contribution is to show that certain practices of computing surrogate references actually can be understood as a form of weak feedback. Coactive learning decouples the learner (performing prediction and updates) from the user (providing feedback in form of an improved translation) so that we can compare different surrogacy modes as different ways of approximate utility maximization. We show experimentally that learning from surrogate "hope" derivations (Chiang, 2012) minimizes regret and TER, thus favoring surrogacy modes that admit an underlying linear model, over "local" updates (Liang et al., 2006) or "oracle" derivations (Sokolov et al., 2013), for which learning does not converge.

It is important to note that the goal of our experiments is not to present improvements of coactive learning over the "optimal" full-information model in terms of standard SMT performance. Instead, our goal is to present experiments that serve as a proof-of-concept of the feasibility of coactive learning from weak feedback for SMT, and to propose a new perspective on standard practices of learning from surrogate translations. The rest of this paper is organized as follows. After a review of related work (Section 2), we present a latent percpetron algorithm and analyze its convergence and generalization properties (Section 3). Our first set of experiments (Section 4.1) confirms our theoretical analysis by showing convergence in regret and TER for learning from weak and strong feedback. Our second set of experiments (Section 4.2) analyzes the relation of different surrogacy modes to minimization of regret and TER.

2 Related Work

Our work builds on the framework of coactive learning, introduced by Shivaswamy and Joachims (2012). We extend their algorithms and proofs to the area of SMT where latent variable models are appropriate, and additionally present generalization guarantees and an online-to-batch conversion. Our theoretical analysis is easily extendable to the full information case of Sun et al. (2013). We also extend our own previous work (Sokolov et al., 2015) with theory and experiments for online-to-batch conversion, and with experiments on coactive learning from surrogate translations.

Online learning has been applied for discriminative training in SMT, based on perceptron-type algorithms (Shen et al. (2004), Watanabe et al. (2006), Liang et al. (2006), Yu et al. (2013), *inter alia*), or large-margin approaches (Tillmann and Zhang (2006), Watanabe et al. (2007), Chiang et al. (2008), Chiang et al. (2009), Chiang (2012), *inter alia*). The latest incarnations are able to handle millions of features and millions of parallel sentences (Simianer et al. (2012), Eidelmann (2012), Watanabe (2012), Green et al. (2013), *inter alia*). Most approaches rely on hidden derivation variables, use some form of surrogate references, and involve n-best lists that change after each update.

Online learning from post-edits has mostly been confined to "simulated post-editing" where independently created human reference translations, or post-edits on the output from similar SMT systems, are used as for online learning (Cesa-Bianchi et al. (2008), López-Salcedo et al. (2012), Martínez-Gómez et al. (2012), Saluja et al. (2012), Saluja and Zhang (2014), *inter alia*). Recent approaches extend online parameter updating by online phrase extraction (Wäschle et al. (2013), Bertoldi et al. (2014), Denkowski et al. (2014), Green et al. (2014), *inter alia*). We exclude dynamic phrase table extension, which has shown to be important in online learning for post-editing, in our theoretical analysis (Denkowski et al., 2014).

Learning from weak feedback is related to binary response-based learning where a meaning representation is "tried out" by iteratively generating system outputs, receiving feedback from world interaction, and updating the model parameters. Such world interaction consists of database access in semantic parsing (Kwiatowski et al. (2013), Berant et al. (2013), or Goldwasser and Roth (2013), *inter alia*). Feedback in response-based learning is given by a user accepting or rejecting system predictions, but not by user corrections.

Lastly, feedback in form of numerical utility values for actions is studied in the frameworks of reinforcement learning (Sutton and Barto, 1998) or in online learning with limited feedback, e.g., multi-armed bandit models (Cesa-Bianchi and Lugosi, 2006). Our framework replaces quantitative feedback with immediate qualitative feedback in form of a structured object that improves upon the utility of the prediction.

3 Coactive Learning for Online Latent Structured Prediction

3.1 Notation and Background

Let $\mathcal{X}$ denote a set of input examples, e.g., sentences, and let $\mathcal{Y}(x)$ denote a set of structured outputs for $x \in \mathcal{X}$, e.g., translations. We define $\mathcal{Y} = \cup_x \mathcal{Y}(x)$. Furthermore, by $\mathcal{H}(x,y)$ we denote a set of possible hidden derivations for a structured output $y \in \mathcal{Y}(x)$, e.g., for phrase-based SMT, the hidden derivation is determined by a phrase segmentation and a phrase alignment between source and target sentences. Every hidden derivation $h \in \mathcal{H}(x,y)$ deterministically identifies an output $y \in \mathcal{Y}(x)$. We define $\mathcal{H} = \cup_{x,y} \mathcal{H}(x,y)$. Let $\phi \colon \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \rightarrow \mathbb{R}^d$ denote a feature function that maps a triplet (x,y,h) to a d-dimensional vector. For phrase-based SMT, we use 14 features, defined by phrase translation probabilities, language model probability, distance-based and lexicalized reordering probabilities, and word and phrase penalty. We assume that the feature function has a bounded radius, i.e. that $\|\phi(x,y,h)\| \leq R$ for all x, y, h. By $\Delta_{h,h'}$ we denote a distance function that is defined for any $h, h' \in \mathcal{H}$, and is used to scale the step size of updates during learning. In our experiments, we use the ordinary Euclidean distance between the feature vectors of derivations. We assume a linear model with fixed parameters w_* such that each input example is mapped to its correct derivation and structured output by using $(y^*, h^*) = \arg\max_{y \in \mathcal{Y}(x), h \in \mathcal{H}(x,y)} {w_*}^\top \phi(x,y,h)$. We define for each given input x, its highest scoring derivation over all outputs $\mathcal{Y}(x)$ such that $h(x;w) = \arg\max_{h' \in \mathcal{H}(x,y)} \max_{y \in \mathcal{Y}(x)} w^\top \phi(x,y,h')$ and the highest scoring derivation for a given output $y \in \mathcal{Y}(x)$ such that $h(x|y;w) = \arg\max_{h' \in \mathcal{H}(x,y)} w^\top \phi(x,y,h')$. In the following theoretical exposition we assume that the $\arg\max$ operation can be computed exactly.

Algorithm 1 Feedback-based Latent Perceptron

```
Initialize $w \leftarrow 0$
for $t = 1, \ldots, T$ do
    Observe $x_t$
    $(y_t, h_t) \leftarrow \arg\max_{(y,h)} w_t^\top \phi(x_t, y, h)$
    Obtain weak feedback $\bar{y}_t$
    if $y_t \neq \bar{y}_t$ then
        $\bar{h}_t \leftarrow \arg\max_h w_t^\top \phi(x_t, \bar{y}_t, h)$
        $w_{t+1} \leftarrow w_t + \Delta_{\bar{h}_t, h_t} \big(\phi(x_t, \bar{y}_t, \bar{h}_t) - \phi(x_t, y_t, h_t)\big)$
```

3.2 Feedback-based Latent Perceptron

We assume an online setting, in which examples are presented one-by-one. The learner observes an input x_t, predicts an output structure y_t, and is presented with feedback $\bar{y}_t$ about its prediction, which is used to make an update to an existing parameter vector. Algorithm 1 is called "Feedback-based Latent Perceptron" to stress the fact that it only uses weak feedback to its predictions for learning, but does not necessarily observe optimal structures as in the full information case (Sun et al., 2013). Learning from full information can be recovered by setting the informativeness parameter α to 1 in Equation (2) below, in which case the feedback structure $\bar{y}_t$ equals the optimal structure y_t^*. Algorithm 1 differs from the algorithm of Shivaswamy and Joachims (2012) by a joint maximization over output structures y and hidden derivations h in prediction (line 4), by choosing a hidden derivation $\bar{h}$ for the feedback structure $\bar{y}$ (line 7), and by the use of the re-scaling factor $\Delta_{\bar{h}_t,h_t}$ in the update (line 8), where $\bar{h}_t = h(x_t|\bar{y}_t;w_t)$ and $h_t = h(x_t;w_t)$ are the derivations of the feedback structure and the prediction at time t, respectively. In our theoretical exposition, we assume that $\bar{y}_t$ is reachable in the search space of possible outputs, that is, $\bar{y}_t \in \mathcal{Y}(x_t)$.

3.3 Feedback of Graded Utility

The key in the theoretical analysis in Shivaswamy and Joachims (2012) is the notion of a linear utility function, determined by parameter vector w_*, that is unknown to the learner:

$$U_h(x,y) = {w_*}^\top \phi(x,y,h).$$

Upon a system prediction, the user approximately maximizes utility, and returns an improved object $\bar{y}_t$ that has higher utility than the predicted y_t s.t.

$$U(x_t,\bar{y}_t) > U(x_t,y_t)$$

where for given $x \in \mathcal{X}$, $y \in \mathcal{Y}(x)$, and $h^* = \arg\max_{h\in\mathcal{H}(x,y)} U_h(x,y)$, we define $U(x,y) = U_{h^*}(x,y)$ and drop the subscript unless $h \neq h^*$. Importantly, the feedback is typically not the optimal structure y_t^* that is defined as

$$y_t^* = \arg\max_{y\in\mathcal{Y}(x_t)} U(x_t,y).$$

While not receiving optimal structures in training, the learning goal is to predict objects with utility close to optimal structures y_t^*. The regret that is suffered by the algorithm when predicting object y_t instead of the optimal object y_t^* is

$$\mathrm{REG}_T = \frac{1}{T}\sum_{t=1}^{T}\big(U(x_t,y_t^*) - U(x_t,y_t)\big). \quad (1)$$

To quantify the amount of information in the weak feedback, Shivaswamy and Joachims (2012) define a notion of *α-informative* feedback, which we generalize as follows for the case of latent derivations. We assume that there exists a derivation $\bar{h}_t$ for the feedback structure $\bar{y}_t$, such that for all predictions y_t, the (re-scaled) utility of the weak feedback $\bar{y}_t$ is higher than the (re-scaled) utility of the prediction y_t by a fraction α of the maximum possible utility range (under the given utility model). Thus $\forall t, \exists\bar{h}_t, \forall h$ and for $\alpha \in (0,1]$:

$$\begin{aligned}\big(U_{\bar{h}_t}(x_t,\bar{y}_t) - U_h(x_t,y_t)\big) \times \Delta_{\bar{h}_t,h}&\\ \geq \alpha\big(U(x_t,y_t^*) - U(x_t,y_t)\big) - \xi_t,& \quad (2)\end{aligned}$$

where $\xi_t \geq 0$ are slack variables allowing for violations of (2) for given α. For slack $\xi_t = 0$, user feedback is called *strictly α-informative*.

3.4 Convergence Analysis

A central theoretical result in learning from weak feedback is an analysis that shows that Algorithm 1 minimizes an upper bound on the average regret (1), despite the fact that optimal structures are not used in learning:

Theorem 1. *Let $D_T = \sum_{t=1}^T \Delta^2_{\bar{h}_t,h_t}$. Then the average regret of the feedback-based latent perceptron can be upper bounded for any $\alpha \in (0,1]$, for any $w_* \in \mathbb{R}^d$:*

$$\mathrm{REG}_T \leq \frac{1}{\alpha T}\sum_{t=1}^{T}\xi_t + \frac{2R\|w_*\|}{\alpha}\frac{\sqrt{D_T}}{T}.$$

A proof for Theorem 1 is similar to the proof of Shivaswamy and Joachims (2012) and the original mistake bound for the perceptron of Novikoff (1962).[1] The theorem can be interpreted as follows: we expect lower average regret for higher values of α; due to the dominant term T, regret will approach the minimum of the accumulated slack (in case feedback structures violate Equation (2)) or 0 (in case of strictly α-informative feedback). The main difference between the above result and the result of Shivaswamy and Joachims (2012) is the term D_T following from the rescaled distance of latent derivations. Their analysis is agnostic of latent derivations, and can be recovered by setting this scaling factor to 1. This yields $D_T = T$, and thus recovers the main factor $\frac{\sqrt{D_T}}{T} = \frac{1}{\sqrt{T}}$ in their regret bound. In our algorithm, penalizing large distances of derivations can help to move derivations h_t closer to $\bar{h}_t$, therefore decreasing D_T as learning proceeds. Thus in case $D_T < T$, our bound is better than the original bound of Shivaswamy and Joachims (2012) for a perceptron without re-scaling. As we will show experimentally, re-scaling leads to a faster convergence in practice.

3.5 Generalization Analysis

Regret bounds measure how good the average prediction of the current model is on the next example in the given sequence, thus it seems plausible that a low regret on a sequence of examples should imply good generalization performance on the entire domain of examples.

[1] Short proofs are provided in the appendix.

Generalization for Online Learning. First we present a generalization bound for the case of online learning on a sequence of random examples, based on generalization bounds for expected average regret as given by Cesa-Bianchi et al. (2004). Let probabilities $\mathbb{P}$ and expectations $\mathbb{E}$ be defined with respect to the fixed unknown underlying distribution according to which all examples are drawn. Furthermore, we bound our loss function $\ell_t = U(x_t, y_t^*) - U(x_t, y_t)$ to $[0, 1]$ by adding a normalization factor $2R||w_*||$ s.t. $\mathrm{REG}_T = \frac{1}{T}\sum_{t=1}^{T}\ell_t$. Plugging the bound on REG_T of Theorem 1 directly into Proposition 1 of Cesa-Bianchi et al. (2004) gives the following theorem:

Theorem 2. *Let $0 < \delta < 1$, and let $x_1, \dots, x_T$ be a sequence of examples that Algorithm 1 observes. Then with probability at least $1 - \delta$,*

$$\mathbb{E}[\mathrm{REG}_T] \leq \frac{1}{\alpha T}\sum_{t=1}^{T}\xi_t + \frac{2R\|w_*\|}{\alpha}\frac{\sqrt{D_T}}{T} + 2||w_*||R\sqrt{\frac{2}{T}\ln\frac{1}{\delta}}.$$

The generalization bound tells us how far the expected average regret $\mathbb{E}[\mathrm{REG}_T]$ (or average risk, in terms of Cesa-Bianchi et al. (2004)) is from the average regret that we actually observe in a specific instantiation of the algorithm.

Generalization for Online-to-Batch Conversion. In practice, perceptron-type algorithms are often applied in a batch learning scenario, i.e., the algorithm is applied for K epochs to a training sample of size T and then used for prediction on an unseen test set (Freund and Schapire, 1999; Collins, 2002). The difference to the online learning scenario is that we treat the multi-epoch algorithm as an empirical risk minimizer that selects a final weight vector $w_{T,K}$ whose expected loss on unseen data we would like to bound. We assume that the algorithm is fed with a sequence of examples $x_1, \dots, x_T$, and at each epoch $k = 1, \dots, K$ it makes a prediction $y_{t,k}$. The correct label is y_t^*. For $k = 1, \dots, K$ and $t = 1, \dots, T$, let $\ell_{t,k} = U(x_t, y_t^*) - U(x_t, y_{t,k})$, and denote by $\Delta_{t,k}$ and $\xi_{t,k}$ the distance at epoch k for example t, and the slack at epoch k for example t, respectively. Finally, we denote by $D_{T,K} = \sum_{t=1}^{T}\Delta_{t,K}^2$, and by $w_{T,K}$ the final weight vector returned after K epochs. We state a condition of convergence[2]:

Condition 1. *Algorithm 1 has converged on training instances $x_1, \dots, x_T$ after K epochs if the predictions on $x_1, \dots, x_T$ using the final weight vector $w_{T,K}$ are the same as the predictions on $x_1, \dots, x_T$ in the Kth epoch.*

Denote by $\mathbb{E}_X(\ell(x))$ the expected loss on unseen data when using $w_{T,K}$ where $\ell(x) = U(x, y^*) - U(x, y')$, $y^* = \arg\max_y U(x, y)$ and $y' = \arg\max_y \max_h w_{T,K}^\top \phi(x, y, h)$. We can now state the following result:

Theorem 3. *Let $0 < \delta < 1$, and let $x_1, \dots, x_T$ be a sample for the multiple-epoch perceptron algorithm such that the algorithm converged on it (Condition 1). Then, with probability at least $1-\delta$, the expected loss of the feedback-based latent perceptron satisfies:*

$$\mathbb{E}_X(\ell(x)) \leq \frac{1}{\alpha T}\sum_{t=1}^{T}\xi_{t,K} + \frac{2R\|w_*\|}{\alpha}\frac{\sqrt{D_{T,K}}}{T} + R\|w_*\|\sqrt{\frac{8\ln\frac{2}{\delta}}{T}}.$$

The theorem can be interpreted as a bound on the generalization error (lefthand-side) by the empirical error (the first two righthand-side terms) and the variance caused by the finite sample (the third term in the theorem). The result follows directly from McDiarmid's concentration inequality.

4 Experiments

We used the LIG corpus[3] which consists of 10,881 tuples of French-English post-edits (Potet et al., 2012). The corpus is a subset of the news-commentary dataset provided at WMT[4] and contains input French sentences, MT outputs, post-edited outputs and English references. To prepare SMT outputs for post-editing, the creators of the corpus used their own WMT10 system (Potet et al., 2010), based on the Moses phrase-based decoder (Koehn et al., 2007) with dense features. We replicated a similar Moses system using the same monolingual and parallel data: a 5-gram language model was estimated with the KenLM toolkit (Heafield, 2011) on `news.en` data (48.65M sentences, 1.13B tokens), pre-processed with the tools from the `cdec` toolkit (Dyer et al., 2010).

[2] This condition is too strong for large datasets. However, we believe that a weaker condition based on ideas from the perceptron cycling theorem (Block and Levin, 1970; Gelfand et al., 2010) should suffice to show a similar bound.

[3] http://www-clips.imag.fr/geod/User/marion.potet/index.php?page=download

[4] http://statmt.org/wmt10/translation-task.html

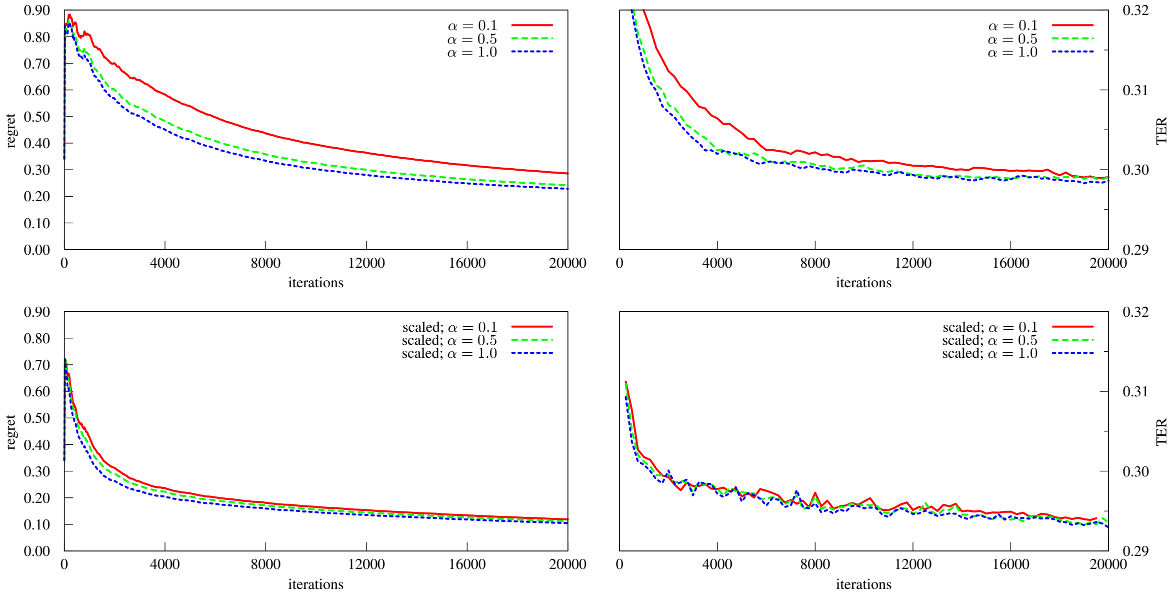

Figure 1: Regret and TER vs. iterations for α-informative feedback ranging from weak ($\alpha = 0.1$) to strong ($\alpha = 1.0$) informativeness, with (lower part) and without re-scaling (upper part).

Parallel data (europarl+news-comm, 1.64M sentences) were similarly pre-processed and aligned with fast_align (Dyer et al., 2013). In all experiments, training is started with the Moses default weights. The size of the n-best list, where used, was set to 1,000. Irrespective of the use of re-scaling in perceptron training, a constant learning rate of 10^{-5} was used for learning from simulated feedback, and 10^{-4} for learning from surrogate translations.

Our experiments on online learning require a random sequence of examples for learning. Following the techniques described in Bertsekas (2011) to generate random sequences for incremental optimization, we compared *cyclic* order (K epochs of T examples in fixed order), *randomized* order (sampling datapoints with replacement), and random *shuffling* of datapoints after each cycle, and found nearly identical regret curves for all three scenarios. In the following, all figures are shown for sequences in the cyclic order, with re-decoding after each update. Furthermore note that in all three definitions of sequence, we never see the fixed optimal feedback y_t^* in training, but instead in general a different feedback structure $\bar{y}_t$ (and a different prediction y_t) every time we see the same input x_t.

4.1 Idealized Weak and Strong Feedback

In a first experiment, we apply Algorithm 1 to user feedback of varying utility grade. The goal of this experiment is to confirm our theoretical analysis by showing convergence in regret for learning from weak and strong feedback. We select feedback of varying grade by directly inspecting the optimal w_*, thus this feedback is idealized. However, the experiment also has a realistic background since we show that α-informative feedback corresponds to improvements under standard evaluation metrics such as lowercased and tokenized TER, and that learning from weak and strong feedback leads to convergence in TER on test data.

	strict ($\xi_t = 0$)	slack ($\xi_t > 0$)
# datapoints	5,725	1,155
TER($\bar{y}_t$) < TER(y_t)	52.17%	32.55%
TER($\bar{y}_t$) = TER(y_t)	23.95%	20.52%
TER($\bar{y}_t$) > TER(y_t)	23.88%	46.93%

Table 1: Improved utility vs. improved TER distance to human post-edits for α-informative feedback $\bar{y}_t$ compared to prediction y_t using default weights at $\alpha = 0.1$.

For this experiment, the post-edit data from the LIG corpus were randomly split into 3 subsets: PE-train (6,881 sentences), PE-dev, and PE-test (2,000 sentences each). PE-train was used for our online learning experiments. PE-test was held out for testing the algorithms' progress on unseen data. PE-dev was used to obtain w_* to define the utility model. This was done by MERT optimization (Och, 2003) towards post-edits under the TER target metric. Note that the goal of our experi-

	% strictly α-informative
local	39.46%
filtered	47.73%
hope	83.30%

Table 2: α-informativeness of surrogacy modes.

ments is not to improve SMT performance over any algorithm that has access to full information to compute w_*. Rather, we want to show that learning from weak feedback leads to convergence in regret with respect to the optimal model, albeit at a slower rate than learning from strong feedback. The feedback data in this experiment were generated by searching the n-best list for translations that are α-informative at $\alpha \in \{0.1, 0.5, 1.0\}$ (with possible non-zero slack). This is achieved by scanning the n-best list output for every input x_t and returning the first $\bar{y}_t \neq y_t$ that satisfies Equation (2).[5] This setting can be thought of as an idealized scenario where a user picks translations from the n-best list that are considered improvements under the optimal w_*.

In order to verify that our notion of graded utility corresponds to a realistic concept of graded translation quality, we compared improvements in utility to improved TER distance to human post-edits. Table 1 shows that for predictions under default weights, we obtain strictly α-informative (for $\alpha = 0.1$) feedback for 5,725 out of 6,881 datapoints in PE-train. These feedback structures improve utility per definition, and they also yield better TER distance to post-edits in the majority of cases. A non-negative slack has to be used in 1,155 datapoins. Here the majority of feedback structures do not improve TER distance.

Convergence results for different learning scenarios are shown in Figure 1. The left upper part of Figure 1 shows average utility regret against iterations for a setup without re-scaling, i.e., setting $\Delta_{\bar{h},h} = 1$ in the definition of α-informative feedback (Equation (2)) and in the update of Algorithm 1 (line 8). As predicted by our regret analysis, higher α leads to faster convergence, but all three curves converge towards a minimal regret. Also, the difference between the curves for $\alpha = 0.1$ and $\alpha = 1.0$ is much smaller than a factor of ten. As expected from the correspondence of α-informative feedback to improvements in TER, similar relations are obtained when plotting TER scores on test data for training from weak feedback at different utility grades. This is shown in the right upper part of Figure 1.

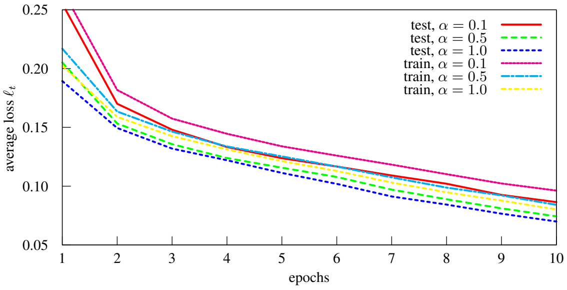

Figure 3: Average loss ℓ_t on heldout and train data.

The left lower part of Figure 1 shows average utility regret plotted against iterations for a setup that uses re-scaling. We define $\Delta_{\bar{h}_t,h}$ by the ℓ_2-distance between the feature vectors $\phi(x_t, \bar{y}_t, \bar{h}_t)$ of the derivation of the feedback structure and the feature vector $\phi(x_t, y_t, h_t)$ of the derivation of the predicted structure. We see that the curves for all grades of feedback converge faster than the corresponding curves for un-scaled feedback shown in the upper part Figure 1. Furthermore, as shown in the right lower part of Figure 1, TER is decreased on test data as well at a faster rate.[6]

Lastly, we present an experimental validation of the online-to-batch application of our algorithm. That is, we would like to evaluate predictions that use the final weight vector $w_{T,K}$ by comparing the generalization error with the empirical error stated in Theorem 3. The standard way to do this is to compare the average loss on heldout data with the the average loss on the training sequence. Figure 3 shows these results for models trained on α-informative feedback of $\alpha \in \{0.1, 0.5, 1.0\}$ for 10 epochs. Similar to the online learning setup, higher α results in faster convergence. Furthermore, curves for training and heldout evaluation converge at the same rate.

4.2 Feedback from Surrogate Translations

In this section, we present experiments on learning from real human post-edits. The goal of this experiment is to investigate whether the stan-

[5] Note that feedback provided in this way might be stronger than required at a particular value of α since for all $\beta \geq \alpha$, strictly β-informative feedback is also strictly α-informative. On the other hand, because of the limited size of the n-best list, we cannot assume strictly α-informative user feedback with zero slack ξ_t. In experiments where updates are only done if feedback is strictly α-informative we found similar convergence behavior.

[6] We also conducted online-to-batch experiments for simulated feedback at $\alpha \in \{0.1, 0.5, 1.0\}$. Similar to the online learning setup, higher α results in faster convergence.

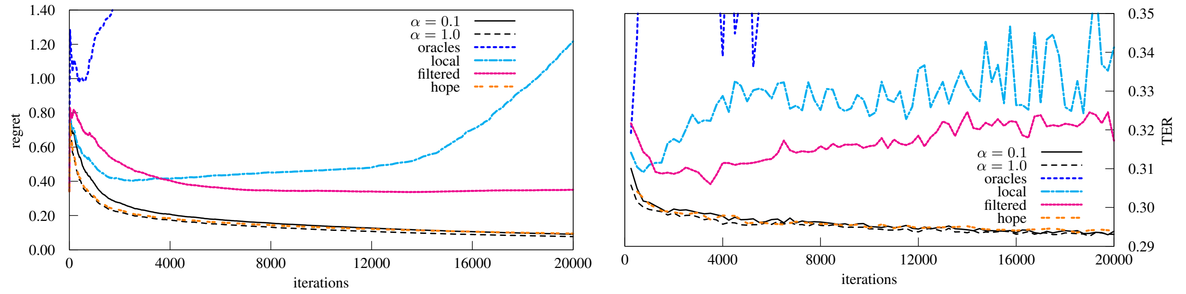

Figure 2: Regret and TER for online learning from `oracles`, `local`, `filtered`, and `hope` surrogates.

dard practices for extracting feedback from observed user post-edits for discriminative SMT can be matched with the modeling assumptions of the coactive learning framework. The customary practice in discriminative learning for SMT is to replace observed user translations by surrogate translations since the former are often not reachable in the search space of the SMT decoder. In our case, only 29% of the post-edits in the LIG-corpus were reachable by the decoder. We compare four heuristics of generating surrogate translations: `oracles` are generated using the lattice oracle approach of Sokolov et al. (2013) which returns the closest path in the decoder search graph as reachable surrogate translation.[7] A `local` surrogate $\tilde{y}$ is chosen from the n-best list of the linear model as the translation that achieves the best TER score with respect to the actual post-edit y: $\tilde{y} = \arg\min_{y' \in n\text{-best}(x_t; w_t)} \text{TER}(y', y)$. This corresponds to the local update mode of Liang et al. (2006). A `filtered` surrogate translation $\tilde{y}$ is found by scanning down the n-best list, and accepting the first translation as feedback that improves TER score with respect to the human post-edit y over the 1-best prediction y_t of the linear model: $\text{TER}(\tilde{y}, y) < \text{TER}(y_t, y)$. Finally, a `hope` surrogate is chosen from the n-best list as the translation that jointly maximizes model score under the linear model and negative TER score with respect to the human post-edit: $\tilde{y} = \arg\max_{y' \in n\text{-best}(x_t; w_t)}(-\text{TER}(y', y) + w_t^\top \phi(x_t, y', h))$. This corresponds to what Chiang (2012) termed "hope derivations". Informally, `oracles` are model-agnostic, as they can pick a surrogate even from outside of the n-best list; `local` is constrained to the n-best list, though still ignoring the ordering according to the linear model; finally, `filtered` and `hope` represent different ways of letting the model score influence the selected surrogate.

As shown in Figure 2, regret and TER decrease with the increased amount of information about the assumed linear model that is induced by the surrogate translations: Learning from `oracle` surrogates does not converge in regret and TER. The `local` surrogates extracted from 1,000-best lists still do not make effective use of the linear model, while `filtered` surrogates enforce an improvement over the prediction under TER towards the human post-edit, and improve convergence in learning. Empirically, convergence is achieved only for `hope` surrogates that jointly maximize negative TER and linear model score, with a convergence behavior that is very similar to learning from weak α-informative feedback at $\alpha \simeq 0.1$. We quantify this in Table 2 where we see that the improvement in TER over the prediction that holds for any `hope` derivation, corresponds to an improvement in α-informativeness: `hope` surrogates are strictly α-informative in 83.3% of the cases in our experiment, whereas we find a correspondence to strict α-informativeness only in 45.74% or 39.46% of the cases for `filtered` and `local` surrogates, respectively.

5 Discussion

We presented a theoretical analysis of online learning for SMT from a coactive learning perspective. This viewpoint allowed us to give regret and generalization bounds for perceptron-style online learners that fall outside the convex optimization scenario because of latent variables and changing feedback structures. We introduced the concept of weak feedback into online learning for SMT, and provided proof-of-concept experiments whose goal was to show that learning from weak feedback converges to minimal regret, albeit at a

[7] While the original algorithm is designed to maximize the BLEU score of the returned path, we tuned its two free parameters to maximize TER.

slower rate than learning from strong feedback. Furthermore, we showed that the SMT standard of learning from surrogate `hope` derivations can to be interpreted as a search for weak improvements under the assumed linear model. This justifies the importance of admitting an underlying linear model in computing surrogate derivations from a coactive learning perspective.

Finally, we hope that our analysis motivates further work in which the idea of learning from weak feedback is taken a step further. For example, our results could perhaps be strengthened by applying richer feature sets or dynamic phrase table extension in experiments on interactive SMT. Our theory would support a new post-editing scenario where users pick translations from the n-best list that they consider improvements over the prediction. Furthermore, it would be interesting to see if "light" post-edits that are better reachable and easier elicitable than "full" post-edits provide a strong enough signal for learning.

Acknowledgments

This research was supported in part by DFG grant RI-2221/2-1 "Grounding Statistical Machine Translation in Perception and Action."

Appendix: Proofs of Theorems

Proof of Theorem 1

Proof. First we bound $w_{T+1}^\top w_{T+1}$ from above:

$$\begin{aligned} w_{T+1}^\top w_{T+1} &= w_T^\top w_T \\ &\quad + 2w_T^\top \big(\phi(x_T,\bar{y}_T,\bar{h}_T) - \phi(x_T,y_T,h_T)\big)\Delta_{\bar{h}_T,h_T} \\ &\quad + \big(\phi(x_T,\bar{y}_T,\bar{h}_T) - \phi(x_T,y_T,h_T)\big)^\top \Delta_{\bar{h}_T,h_T} \\ &\qquad \big(\phi(x_T,\bar{y}_T,\bar{h}_T) - \phi(x_T,y_T,h_T)\big)\Delta_{\bar{h}_T,h_T} \\ &\le w_T^\top w_T + 4R^2\Delta^2_{\bar{h}_T,h_T} \le 4R^2 D_T. \end{aligned} \tag{3}$$

The first equality uses the update rule from Algorithm 1. The second uses the fact that $w_T^\top(\phi(x_T,\bar{y}_T,\bar{h}_T) - \phi(x_T,y_T,h_T)) \le 0$ by definition of (y_T,h_T) in Algorithm 1. By assumption $\|\phi(x,y,h)\| \le R, \forall x,y,h$ and by the triangle inequality, $\|\phi(x,y,h) - \phi(x,y',h')\| \le \|\phi(x,y,h)\| + \|\phi(x,y',h')\| \le 2R$. Finally, $D_T = \sum_{t=1}^T \Delta^2_{\bar{h}_t,h_t}$ by definition, and the last inequality follows by induction.

The connection to average regret is as follows:

$$\begin{aligned} w_{T+1}^\top w_* &= w_T^\top w_* \\ &\quad + \Delta_{\bar{h}_T,h_T}\big(\phi(x_T,\bar{y}_T,\bar{h}_T)) - \phi(x_T,y_T,h_T)\big)^\top w_* \\ &= \sum_{t=1}^T \Delta_{\bar{h}_t,h_t}\big(\phi(x_t,\bar{y}_t,\bar{h}_t) - \phi(x_t,y_t,h_t)\big)^\top w_* \\ &= \sum_{t=1}^T \Delta_{\bar{h}_t,h_t}\big(U_{\bar{h}_t}(x_t,\bar{y}_t) - U_{h_t}(x_t,y_t)\big). \end{aligned} \tag{4}$$

The first equality again uses the update rule from Algorithm 1. The second follows by induction. The last equality applies the definition of utility.

Next we upper bound the utility difference:

$$\begin{aligned} &\sum_{t=1}^T \Delta_{\bar{h}_t,h_t}\big(U_{\bar{h}_t}(x_t,\bar{y}_t) - U_{h_t}(x_t,y_t)\big) \\ &\qquad \le \|w_*\|\|w_{T+1}\| \le \|w_*\| 2R\sqrt{D_T}. \end{aligned} \tag{5}$$

The first inequality follows from applying the Cauchy-Schwartz inequality $w_{T+1}^\top w_* \le \|w_*\|\|w_{T+1}\|$ to Equation (4). The seond follows from applying Equation (3) to $\|w_{T+1}\| = \sqrt{w_{T+1}^\top w_{T+1}}$.

The final result is obtained simply by lower bounding Equation (5) using the assumption in Equation (2).

$$\begin{aligned} &\|w_*\| 2R\sqrt{D_T} \\ &\quad \ge \sum_{t=1}^T \Delta_{\bar{h}_t,h_t}\big(U_{\bar{h}_t}(x_t,\bar{y}_t) - U_{h_t}(x_t,y_t)\big) \\ &\quad \ge \alpha \sum_{t=1}^T \big(U(x_t,y_t^*) - U(x_t,y_t)\big) - \sum_{t=1}^T \xi_t \\ &\quad = \alpha\, T\, \mathrm{REG}_T - \sum_{t=1}^T \xi_t. \end{aligned}$$

□

Proof of Theorem 3

Proof. The theorem can be shown by an application of McDiarmid's concentration inequality:

Theorem 4 (McDiarmid, 1989)**.** *Let $Z_1,\dots,Z_m$ be a set of random variables taking value in a set $\mathcal{Z}$. Further, let $f\colon \mathcal{Z}^m \to \mathbb{R}$ be a function that satisfies for all i and $z_1,\dots,z_m,z_i' \in \mathcal{Z}$:*

$$\begin{aligned} &|f(z_1,\dots,z_i,\dots,z_m) \\ &\qquad - f(z_1,\dots,z_i',\dots,z_m)| \le c, \end{aligned} \tag{6}$$

for some c. Then for all $\epsilon > 0$,

$$\mathbb{P}(|f - \mathbb{E}(f)| > \epsilon) \le 2\exp(-\frac{2\epsilon^2}{mc^2}). \tag{7}$$

Let f be the average loss for predicting y_t on example x_t in epoch K: $f(x_1,\dots,x_T) = \mathrm{REG}_{T,K} = \frac{1}{T}\sum_{t=1}^T \ell_{t,K}$. Because of the convergence condition (Condition 1), $\ell_{t,K} = \ell(x_t)$. The expectation of f is $\mathbb{E}(f) = \frac{1}{T}\sum_{t=1}^T \mathbb{E}[\ell_{t,k}] = \frac{1}{T}\sum_{t=1}^T \mathbb{E}[\ell(x_t)] = \mathbb{E}_X(\ell(x))$.

The first and second term on the righthand-side of Theorem 3 follow from upper bounding REG_T in the Kth epochs, using Theorem 1. The third term is derived by calculating c in Equation (6) as follows:

$$\begin{aligned} &|f(x_1,\dots,x_t,\dots,x_T) - f(x_1,\dots,x_t',\dots,x_T)| \\ &\quad = |\frac{1}{T}\sum_{t=1}^T \ell_{t,K} - \frac{1}{T}\sum_{t=1}^T \ell_{t,K}'| = |\frac{1}{T}\sum_{t=1}^T \big(\ell_{t,K} - \ell_{t,K}'\big)| \\ &\quad \le \frac{1}{T}\sum_{t=1}^T \big(|\ell_{t,k}| + |\ell_{t,K}'|\big) \le \frac{4R\|w_*\|}{T} = c. \end{aligned}$$

The first inequality uses the triangle inequality; the second uses the upper bound $|\ell_{t,k}| \le 2R\|w_*\|$. Setting the righthand-side of Equation (7) to at least δ and solving for ϵ, using c, concludes the proof. □

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, Seattle, WA.

Nicola Bertoldi, Patrick Simianer, Mauro Cettolo, Katharina Wäschle, Marcello Federico, and Stefan Riezler. 2014. Online adaptation to post-edits for phrase-based statistical machine translation. *Machine Translation*, 29:309–339.

Dimitri P. Bertsekas. 2011. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. In Suvrit Sra, Sebastian Nowozin, and Stephen J. Wright, editors, *Optimization for Machine Learning*. MIT Press.

Henry D. Block and Simon A. Levin. 1970. On the boundedness of an iterative procedure for solving a system of linear inequalities. *Proceedings of the American Mathematical Society*, 26(2):229–235.

Leon Bottou and Olivier Bousquet. 2004. Large scale online learning. In *NIPS*, Vancouver, Canada.

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

Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. 2004. On the generalization ablility of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057.

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

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

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *NAACL*, Boulder, CO.

David Chiang. 2012. Hope and fear for discriminative training of statistical translation models. *Journal of Machine Learning Research*, 12:1159–1187.

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

Ronan Collobert, Fabian Sinz, Jason Weston, and Leon Bottou. 2006. Trading convexity for scalability. In *ICML*, Pittsburgh, PA.

Michael Denkowski, Chris Dyer, and Alon Lavie. 2014. Learning from post-editing: Online model adaptation for statistical machine translation. In *EACL*, Gothenburg, Sweden.

Chuong B. Do, Quoc Le, and Choon Hui Teo. 2008. Tighter bounds for structured estimation. In *NIPS*, Vancouver, Canada.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Türe, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *ACL*, Uppsala, Sweden.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *NAACL*, Atlanta, GA.

Vladimir Eidelmann. 2012. Optimization strategies for online large-margin learning in machine translation. In *WMT*, Montreal, Canada.

Yoav Freund and Robert E. Schapire. 1999. Large margin classification using the perceptron algorithm. *Journal of Machine Learning Research*, 37:277–296.

Andrew E. Gelfand, Yutian Chen, Max Welling, and Laurens van der Maaten. 2010. On herding and the perceptron cycling theorem. In *NIPS*, Vancouver, Canada.

Kevin Gimpel and Noah A. Smith. 2012. Structured ramp loss minimization for machine translation. In *NAACL*, Montreal, Canada.

Dan Goldwasser and Dan Roth. 2013. Learning from natural instructions. *Machine Learning*, 94(2):205–232.

Spence Green, Jeffrey Heer, and Christopher D. Manning. 2013. The efficacy of human post-editing for language translation. In *CHI*, Paris, France.

Spence Green, Sida I. Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D. Manning. 2014. Human effort and machine learnability in computer aided translation. In *EMNLP*, Doha, Qatar.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *WMT*, Edinburgh, UK.

Philipp Koehn, 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*, Prague, Czech Republic.

Tom Kwiatowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *EMNLP*, Seattle, WA.

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

Francisco-Javier López-Salcedo, Germán Sanchis-Trilles, and Francisco Casacuberta. 2012. Online learning of log-linear weights in interactive machine translation. In *IberSpeech*, Madrid, Spain.

Pascual Martínez-Gómez, Germán Sanchis-Trilles, and Francisco Casacuberta. 2012. Online adaptation strategies for statistical machine translation in post-editing scenarios. *Pattern Recognition*, 45(9):3193–3202.

David McAllester and Joseph Keshet. 2011. Generalization bounds and consistency for latent structural probit and ramp loss. In *NIPS*, Granada, Spain.

Colin McDiarmid. 1989. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188.

Albert B.J. Novikoff. 1962. On convergence proofs on perceptrons. *Symposium on the Mathematical Theory of Automata*, 12:615–622.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *NAACL*, Edmonton, Canada.

Marion Potet, Laurent Besacier, and Hervé Blanchon. 2010. The LIG machine translation system for WMT 2010. In *WMT*, Upsala, Sweden.

Marion Potet, Emanuelle Esperança-Rodier, Laurent Besacier, and Hervé Blanchon. 2012. Collection of a large database of French-English SMT output corrections. In *LREC*, Istanbul, Turkey.

Avneesh Saluja and Ying Zhang. 2014. Online discriminative learning for machine translation with binary-valued feedback. *Machine Translation*, 28:69–90.

Avneesh Saluja, Ian Lane, and Ying Zhang. 2012. Machine translation with binary feedback: A large-margin approach. In *AMTA*, San Diego, CA.

Shai Shalev-Shwartz. 2012. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *NAACL*, Boston, MA.

Pannaga Shivaswamy and Thorsten Joachims. 2012. Online structured prediction via coactive learning. In *ICML*, Edinburgh, UK.

Patrick Simianer, Stefan Riezler, and Chris Dyer. 2012. Joint feature selection in distributed stochastic learning for large-scale discriminative training in SMT. In *ACL*, Jeju, Korea.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *AMTA*, Cambridge, MA.

Artem Sokolov, Guillaume Wisniewski, and François Yvon. 2013. Lattice BLEU oracles in machine translation. *Transactions on Speech and Language Processing*, 10(4):18.

Artem Sokolov, Stefan Riezler, and Shay B. Cohen. 2015. Coactive learning for interactive machine translation. In *ICML Workshop on Machine Learning for Interactive Systems (MLIS)*, Lille, France.

Xu Sun, Takuya Matsuzaki, and Wenjie Li. 2013. Latent structured perceptrons for large scale learning with hidden information. *IEEE Transactions on Knowledge and Data Engineering*, 25(9):2064–2075.

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning. An Introduction*. The MIT Press.

Christoph Tillmann and Tong Zhang. 2006. A discriminative global training algorithm for statistical MT. In *COLING-ACL*, Sydney, Australia.

Katharina Wäschle, Patrick Simianer, Nicola Bertoldi, Stefan Riezler, and Marcello Federico. 2013. Generative and discriminative methods for online adaptation in SMT. In *MT Summit*, Nice, France.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2006. NTT statistical machine translation for IWSLT 2006. In *IWSLT*, Kyoto, Japan.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *EMNLP*, Prague, Czech Republic.

Taro Watanabe. 2012. Optimized online rank learning for machine translation. In *NAACL*, Montreal, Canada.

Heng Yu, Liang Huang, Haitao Mi, and Kai Zhao. 2013. Max-violation perceptron and forced decoding for scalable MT training. In *EMNLP*, Seattle, WA.

Alan Yuille and Anand Rangarajan. 2003. The concave-convex procedure. *Neural Computation*, 15:915–936.