Learning from Post-Editing: Online Model Adaptation for Statistical Machine Translation

Michael Denkowski   Chris Dyer   Alon Lavie
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
{mdenkows,cdyer,alavie}@cs.cmu.edu

Abstract

Using machine translation output as a starting point for human translation has become an increasingly common application of MT. We propose and evaluate three computationally efficient online methods for updating statistical MT systems in a scenario where post-edited MT output is constantly being returned to the system: (1) adding new rules to the translation model from the post-edited content, (2) updating a Bayesian language model of the target language that is used by the MT system, and (3) updating the MT system's discriminative parameters with a MIRA step. Individually, these techniques can substantially improve MT quality, even over strong baselines. Moreover, we see super-additive improvements when all three techniques are used in tandem.

1 Introduction

Using machine translation outputs as a starting point for human translators is becoming increasingly common and is now arguably one of the most commercially important applications of MT. Considerable evidence has accumulated showing that human translators are more productive and accurate when *post-editing* MT output than when translating from scratch (Guerberof, 2009; Carl et al., 2011; Koehn, 2012; Zhechev, 2012, *inter alia*). An important (if unsurprising) insight from prior research in this area is that translators become more productive as MT quality improves (Tatsumi, 2009). While general improvements to MT continue to lead to further productivity gains, we explore how MT quality can be improved specifically in an online post-editing scenario in which sentence-level MT outputs are constantly being presented to human experts, edited, and then returned to the system for immediate learning. This task is challenging in two regards. First, from a technical perspective, post-edited outputs must be processed rapidly: a productive post-editor cannot wait for a standard batch MT training pipeline to be rerun after each sentence is corrected! Second, from a methodological perspective, it is expensive to run many human subject experiments, in particular when the human subjects must have translation expertise. We therefore use a **simulated post-editing paradigm** in which either non-post-edited reference translations or manually post-edited translations from a similar MT system are used in lieu of human post-editors (§2). This paradigm allows us to efficiently develop and evaluate systems that can go on to function in real-time post-editing scenarios without modification.

We present and evaluate three online methods for improving translation models using feedback from editors: adding new translations rules to the translation grammar (§3), updating a Bayesian language model with observations of the post-edited output (§4), and using an online discriminative parameter update to minimize model error (§5). These techniques are computationally efficient and make minimal use of approximation or heuristics, handling initial and incremental data in a uniform way. We evaluate these techniques in a variety of language and data scenarios that mimic the demands of real-world translation tasks. Compared to a competitive baseline, we show substantial improvement from updating the translation grammar or language model independently and super-additive gains from combining these techniques with a MIRA update (§6). We then discuss how our techniques relate to prior work (§7) and conclude (§8).

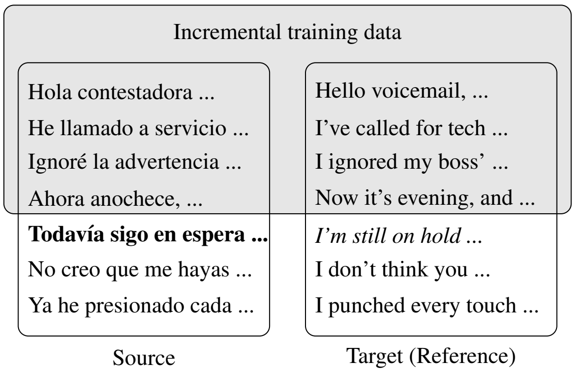
2 Simulated Post-Editing Paradigm

In post-editing scenarios, humans continuously edit machine translation outputs into production-quality translations, providing an additional, con-

Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 395–404,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

stant stream of data absent in batch translation. This data consists of highly domain-relevant reference translations that are minimally different from MT outputs, making them ideal for learning. However, true post-editing data is infeasible to collect during system development and internal testing as standard MT pipelines require tens of thousands of sentences to be translated with low latency. To address this problem, Hardt and Elming (2010) formulate the task of simulated post-editing, wherein pre-generated reference translations are used as a stand-in for actual post-editing. This approximation is equivalent to the case where humans edit each translation hypothesis to be identical to the reference rather than simply correcting the MT output to be grammatical and meaning-equivalent to the source. Our work uses this approximation for tuning and evaluation. We also introduce a more accurate approximation wherein MT output from the target system (or a similar system) is post-edited in advance, creating "offline" post-edited data that is similar to expected system outputs and should thus minimize unnecessary edits. An experiment in §6.4 compares the two approximations.

In our simulated post-editing tasks, decoding (for both the test corpus and each pass over the development corpus during optimization) begins with baseline models trained on standard bilingual and monolingual data. After each sentence is translated, the following take place in order: First, MIRA uses the new source–reference pair to update weights for the current models. Second, the source is aligned to the reference and used to update the translation grammar. Third, the reference is added to the Bayesian language model. As sentences are translated, the models gain valuable context information, allowing them to zero in on the target document and translator. Context is reset at the start of each development or test corpus.[1] This setup, which allows a uniform approach to tuning and decoding, is visualized in Figure 1.



Source	Target (Reference)
Hola contestadora ...	Hello voicemail, ...
He llamado a servicio ...	I've called for tech ...
Ignoré la advertencia ...	I ignored my boss' ...
Ahora anochece, ...	Now it's evening, and ...
Todavía sigo en espera ...	*I'm still on hold ...*
No creo que me hayas ...	I don't think you ...
Ya he presionado cada ...	I punched every touch ...

Figure 1: Context when translating an input sentence (bold) with simulated post-editing. Previous sentences and references (shaded) are added to the training data. *After* the current sentence is translated, it is aligned to the reference (italic) and added to the context for the next sentence.

3 Translation Grammar Adaptation

Translation models (either phrase tables or synchronous grammars) are typically generated offline from large bilingual text. This is reasonable in scenarios where available training data is fixed over long periods of time. However, this approach does not allow adding new data without repeating model estimation in its entirety, which may take hours or days. In this section, we describe a simple technique for incorporating new bilingual training data as soon as it is available. Our approach is an extension of the on-demand grammar extractor described by Lopez (2008a). We extend the work initially designed for on-the-fly grammar extraction from static data (to mitigate the expense of storing large translation grammars), to specifically handle incremental data from post-editing.

3.1 Suffix Array Grammar Extraction

Lopez (2008a) introduces an alternative to traditional model estimation for hierarchical phrase-based statistical machine translation (Chiang, 2007). Rather than estimating a single grammar from all training data, the aligned bitext is indexed using a source-side suffix array (Manber and Myers, 1993). When an input sentence is to be translated, a grammar extraction program *samples* instances of aligned phrase pairs from the suffix array that match the source side of the sentence. Using statistics from these samples rather than the entire bitext, a sentence-specific grammar is rapidly generated. In addition to speed gains from sampling, indexing the source side of the bitext facilitates a more powerful feature set. Rules in on-demand grammars are generated using a sample $\mathcal{S}$ for each source phrase f in the input sentence. The sample, containing pairs $\langle f, e \rangle$, is used to calculate the following statistics:

[1] Initial experiments show this to outperform resetting models on more fine-grained document boundaries, although further investigation is warranted.

Feature	Baseline	Adaptive
coherent $p(e\|f)$	$\frac{\mathcal{C}_{\mathcal{S}}(f,e)}{\|\mathcal{S}\|}$	$\frac{\mathcal{C}_{\mathcal{S}}(f,e)+\mathcal{C}_{\mathcal{L}}(f,e)}{\|\mathcal{S}\|+\|\mathcal{L}\|}$
sample size	$\|\mathcal{S}\|$	$\|\mathcal{S}\|+\|\mathcal{L}\|$
co-occurrence $\langle f,e\rangle$	$\mathcal{C}_{\mathcal{S}}(f,e)$	$\mathcal{C}_{\mathcal{S}}(f,e)$+$\mathcal{C}_{\mathcal{L}}(f,e)$
singleton f	$\mathcal{C}_{\mathcal{S}}(f)$ $=1$	$\mathcal{C}_{\mathcal{S}}(f)+\mathcal{C}_{\mathcal{L}}(f)=$ 1
singleton $\langle f,e\rangle$	$\mathcal{C}_{\mathcal{S}}(f,e)$ $=1$	$\mathcal{C}_{\mathcal{S}}(f,e)+\mathcal{C}_{\mathcal{L}}(f,e)$ $=1$
post-edit support $\langle f,e\rangle$	0	$\mathcal{C}_{\mathcal{L}}(f,e)>0$

Table 1: Phrase feature definitions for baseline and adaptive translation models.

- $\mathcal{C}_{\mathcal{S}}(f,e)$: count of instances in $\mathcal{S}$ where f aligns to e (phrase co-occurrence count).
- $\mathcal{C}_{\mathcal{S}}(f)$: count of instances in $\mathcal{S}$ where f aligns to any target phrase.
- $|\mathcal{S}|$: total number of instances in $\mathcal{S}$, equal to number of occurrences of f in training data, capped by the sample size limit.

These statistics are used to instantiate translation rules $X \rightarrow \langle f,e\rangle$ and calculate scores for the phrase feature set shown in the "Baseline" column of Table 1. Notably, the coherent phrase translation probability that conditions on f occurring in the data ($|\mathcal{S}|$) rather than f being extracted as part of a phrase pair ($\mathcal{C}_{\mathcal{S}}(f)$) is shown by Lopez (2008b) to yield significant improvement over the traditional translation probability.

3.2 Online Grammar Extraction

When a human translator post-edits MT output, a new bilingual sentence pair is created. However, in typical settings, it can be weeks or months before these training instances are incorporated into bilingual data and models retrained. Our extension to on-demand grammar extraction incorporates these new training instances into the model immediately. In addition to a static suffix array that indexes initial data, our system maintains a dynamic lookup table. Each new sentence pair is word-aligned with the model estimated from the initial data (a process often called forced alignment). This makes a generally insignificant approximation with respect to the original alignment model. Extractable phrase pairs are stored in the lookup table and phrase occurrences are counted on the source side. When subsequent grammars are extracted, the suffix array sample $\mathcal{S}$ for each f is accompanied by an exhaustive lookup $\mathcal{L}$ from the lookup table. Matching statistics are calculated from $\mathcal{L}$:

- $\mathcal{C}_{\mathcal{L}}(f,e)$: count of instances in $\mathcal{L}$ where f aligns to e.
- $\mathcal{C}_{\mathcal{L}}(f)$: count of instances in $\mathcal{L}$ where f aligns to any target phrase.
- $|\mathcal{L}|$: total number of instances of f in post-editing data (no size limit).

We use combined statistics from $\mathcal{S}$ and $\mathcal{L}$ to calculate scores for the "Adaptive" feature set defined in Table 1. In addition to updating existing features, we introduce a new indicator feature that identifies rules supported by post-editor feedback. Further, our approach allows us to extract rules that encode translations (phrase mappings and reorderings) *only* observed in the incremental post-editing data. This process, which can be seen as influencing the distribution from which grammars are sampled over time, produces comparable results to the infeasible process of rebuilding the translation model after every sentence is translated with the added benefit of allowing an optimizer to learn a weight for the post-edited data via the post-edit support feature. The simple aggregation of statistics allows our model to handle initial and incremental data in a formally consistent way. Further, any additional features that can be calculated on a suffix array sample can be matched by an incremental data lookup, making our translation model a viable platform for further exploration in online learning for MT.

4 Language Model Adaptation

Adapting language models in an online manner based on the content they are generating has long been seen as a promising technique for improving automatic speech recognition and machine translation (Kuhn and de Mori, 1990; Zhao et al., 2004; Sanchis-Trilles, 2012, *inter alia*). The post-editing scenario we are considering simplifies this process somewhat since rather than only having a posterior distribution over machine-generated outputs (any of which may be ungrammatical), the outputs, once edited by human translators, may be presumed to be grammatical.

We thus take a novel approach to language model adaptation, building on recent work showing that state-of-the-art language models can be inferred as the posterior predictive distribution of a Bayesian language model with hierarchical Pitman-Yor process priors, conditioned on the training corpus (Teh, 2006). The Bayesian formulation provides a natural way to incorporate progressively more data: by updating the posterior distribution given subsequent observations. Furthermore, the nonparametric nature of the model means that the model is well suited to potentially unbounded growth of vocabulary. Unfortunately, in general, Bayesian techniques are computationally difficult to work with. However, hierarchical Pitman-Yor process language models (HPYPLMs) are convenient in this regard since (1) inference can be carried out efficiently in a convenient collapsed representation (the "Chinese restaurant franchise") and (2) the posterior predictive distribution from a single sample provides a high quality language model.

We thus use the following procedure. Using the target side of the bitext as observations, we run the Gibbs sampling procedure described by Teh (2006) for 100 iterations in a 3-gram HPYPLM. The inferred "seating configuration" defines a posterior predictive distribution over words in 2-gram contexts (as with any 3-gram LM) as well as a posterior distribution over how the model will generate subsequent observations. We use the former as a language model component of a translation model. And, as post-edited sentences become available, we add their n-grams to the model using the later. We do not run any Gibbs sampling. Just updating the language model in this way, we obtain the results shown in Table 2 for the experimental conditions described in §6.

Spanish–English	*WMT10*	WMT11	TED1	TED2
HPYPLM	*25.5*	24.8	29.4	26.6
+data	***25.8***	**25.2**	**29.5**	**27.0**
English–Spanish	*WMT10*	WMT11	TED1	TED2
HPYPLM	*25.1*	26.8	26.0	24.3
+data	***25.4***	**27.2**	**26.2**	**25.0**
Arabic–English	*MT08*	MT09	TED1	TED2
HPYPLM	*19.3*	24.7	9.5	10.0
+data	***19.6***	**24.9**	**9.8**	**10.5**

Table 2: BLEU scores for systems with trigram HPYPLM (no large language model), with and without incremental updates from simulated post-editing data. Scores are averages over 3 optimizer runs. Bold scores indicate statistically significant improvement. Tuning set scores are italicized.

5 Learning Feature Weights

MT system parameter optimization (learning feature weights for the decoder) is also typically conducted as a batch process. Discriminative learning techniques such as minimum error rate training (Och, 2003) are used to find feature weights that maximize automatic metric score on a small development corpus. The resulting weight vector is then used to decode given input sentences. Using this approach with post-editing tasks presents two major issues. First, reference translation are only considered after all sentences are translated, a mismatch with post-editing where references are available incrementally. Second, despite the fact that adaptive feature sets become more powerful as post-editing data increases, an optimizer must learn a single corpus-level weight for each feature. This forces an averaging effect that can lead to decoding individual sentences with suboptimal weights. We address the first issue by using reference translations to simulate post-editing (Hardt and Elming, 2010) at tuning time and the second by using a version of the margin-infused relaxed algorithm (Crammer et al., 2006; Eidelman, 2012) to make online parameter updates during decoding. The result is a consistent approach to tuning and decoding that brings out the potential of adaptive models.

5.1 Parameter Optimization

In order to make our decoding process fully consistent with tuning, we introduce an online discriminative parameter update that allows our adaptive translation and language models be weighted appropriately as more data is available. This requires an optimization algorithm that can function as an online learner during decoding as well as a batch optimizer during tuning. Popular optimizers such as MERT (Och, 2003) and pairwise rank optimization (Hopkins and May, 2011) cannot be used due to their reliance on corpus-level optimization. We select the cutting-plane variant of the margin-infused relaxed algorithm (Chiang, 2012; Crammer et al., 2006) with additional extensions described by Eidelman (2012). MIRA is an online large-margin learner that makes a parameter update after each model prediction with the objective of choosing the correct output over the incorrect output by a margin at least as large as the cost of predicting the incorrect output. Applied to MT system optimization on a development corpus, MIRA proceeds as follows. The MT system generates a list of the k best translations for a single input sentence. From the list, a "hope" hypothesis is selected as a translation with both high model score and high automatic metric score. A "fear" hypothesis is selected as a translation with high model score but low metric score. Parameters are updated away from the fear hypothesis, toward the hope hypothesis, and the system processes the next input sentence. This process continues for a set number of passes over the development corpus. All adaptive systems used in our work are optimized with this variant of MIRA using the parameter settings described by Eidelman (2012). For each pass over the data, translation and language models have incremental access to reference translations (simulated post-editing data) as input sentences are translated. Translation and language models reset to using background data only at the beginning of each MIRA iteration.[2]

5.2 Online Parameter Updates

Our optimization strategy allows us to treat decoding as if it were simply the next iteration of MIRA (or alternatively that MIRA makes a single pass over an input corpus that consists of the development data concatenated n times followed by unseen input data). After each sentence is translated, a reference translation (resulting from actual human post-editing in production or simulated post-editing for our experiments) is provided to the models and MIRA makes a parameter update. In the only departure from our optimization setup, we decrease the maximum step size for MIRA (described in §6.2), effectively increasing regularization strength. This allows us to prefer small adjustments to already optimized decoding parameters over the large changes needed during tuning. It is also important to note that by using MIRA for updating weights during both tuning and decoding, we avoid scaling issues between multiple optimizers (such as when tuning with MERT and updating with a passive-aggressive algorithm).

6 Experiments

We evaluate our online extensions to standard machine translation systems in a series of simulated post-editing experiments that cover high-traffic languages and challenging domains. We show incremental improvement from our adaptive models and significantly larger gains when pairing our models with an online parameter update. We finally validate our adaptive system on actual post-edited data.

Spanish–English	*WMT10*	WMT11	TED1	TED2
Base MERT	*29.1*	27.9	32.8	29.6
Base MIRA	*29.2*	28.0	32.7	29.7
G	*29.8*	28.3	34.2	30.7
L	*29.2*	28.1	33.0	29.8
M	*29.2*	28.1	33.1	29.8
G+L+M	***30.0***	**28.8**	**35.2**	**31.3**

English–Spanish	*WMT10*	WMT11	TED1	TED2
Base MERT	*27.8*	29.4	26.5	25.7
Base MIRA	*27.7*	29.6	26.8	26.7
G	*28.1*	29.8	27.9	27.5
L	*27.9*	29.7	26.8	26.5
M	*27.9*	29.7	27.2	26.6
G+L+M	***28.4***	**30.4**	**28.6**	**27.9**

Arabic–English	*MT08*	MT09	TED1	TED2
Base MERT	*21.5*	25.0	10.4	10.5
Base MIRA	*21.2*	25.9	10.6	10.9
G	*21.8*	26.2	11.0	11.7
L	*20.6*	25.7	10.6	10.9
M	*21.3*	25.7	10.8	11.0
G+L+M	***21.8***	**26.5**	**11.4**	**11.8**

Table 3: BLEU scores for baseline and adaptive systems. Scores are averages over three optimizer runs. Highest scores are bold and tuning set scores are italicized. All fully adaptive systems (G+L+M) show statistically significant improvement over both MERT and MIRA baselines.

6.1 Data

We conduct a series of simulated post-editing experiments in three full scale language scenarios: Spanish–English, English–Spanish, and Arabic–English. Spanish–English and English–Spanish systems are trained on the 2012 NAACL WMT (Callison-Burch et al., 2012) constrained resources (2 million bilingual sentences, 300 million words of monolingual Spanish, and 1.1 billion words of monolingual English). Arabic–English systems are trained on the 2012 NIST OpenMT (Przybocki, 2012) constrained bilingual resources plus a selection from the English Gigaword corpus (Parker et al., 2011) (5 million bilingual sentences and 650 million words of monolingual En-

[2] Resetting translation and language models prevents contamination. If models retained state from previous passes over the development set, they would include data for input sentences before they were translated, rather than after as in post-editing.

glish). We tune and evaluate on standard news sets: WMT10 and WMT11 for Spanish–English and English–Spanish, and MT08 and MT09 for Arabic–English. To simulate real-world post editing where one translator works on a document at a time, we use only one of the four available reference translation sets for MT08 and MT09.

We also evaluate on a blind domain adaptation scenario that mimics the demands placed on MT systems in real-world translation tasks. The Web Inventory of Transcribed and Translated Talks (WIT^3) corpus (Cettolo et al., 2012) makes transcriptions of TED talks[3] available in several languages, including English, Spanish, and Arabic. For each language pair, we select two sets of 10 talk transcripts each (2000-3000 sentences) as blind evaluation sets. These sets consist of spoken language covering a broad range of topics. Systems have no access to any training or development data in this domain prior to translation.

6.2 Translation Systems

For each language scenario, we first construct a competitive baseline system. Bilingual data is word aligned using the model described by Dyer et al. (2013) and suffix array-backed translation grammars are extracted using the method described by Lopez (2008a). We add the standard lexical and derivation features[4] from Lopez (2008b) and Dyer et al. (2010). An unpruned, modified Kneser-Ney-smoothed 4-gram language model is estimated using the KenLM toolkit (Heafield et al., 2013). Feature weights are optimized using the lattice-based variant of MERT (Macherey et al., 2008; Och, 2003) on either WMT10 or MT08. Evaluation sets are translated using the `cdec` decoder (Dyer et al., 2010) and evaluated with the BLEU metric (Papineni et al., 2002). These results are listed as “Base MERT” in Table 3. To establish a baseline for our adaptive systems, we tune the same baseline system using cutting-plane MIRA with 500-best lists, the pseudo-document approximation described by Eidelman (2012), and a maximum update size of 0.01. We begin with uniform weights and make 20 passes over the development corpus. Results for this system are listed as “Base MIRA”.

To evaluate the impact of each online model adaptation technique, we report the results for the following systems in Table 3:

- G: Baseline MIRA system with online grammar extraction, including incrementally updating existing phrase features plus an additional indicator feature for post-edit support.
- L: Baseline MIRA with a trigram hierarchical Pitman-Yor process language model that is incrementally updated, including a separate out-of-vocabulary feature.
- M: Baseline MIRA with online feature weight updates from cutting-plane MIRA.

Finally, we report results for a fully adaptive system that includes online grammar, language model, and feature weight updates. This system is reported as “G+L+M”. To account for optimizer instability, all systems are tuned (consisting of running either MERT or MIRA) and evaluated 3 times. We report average scores over optimizer runs and conduct statistical significance tests using the methods described by Clark et al. (2011).

	News		TED Talks	
	New	Supp	New	Supp
Spanish–English	15%	19%	14%	18%
English–Spanish	12%	16%	9%	13%
Arabic–English	9%	12%	23%	28%

Table 5: Percentages of new rules (only seen in incremental data) and post-edit supported rules (Rules from all data for which the “post-edit support $\langle f, e\rangle$” feature fires) in grammars by domain.

6.3 Results

Our simulated translation post-editing experiments are summarized in Table 3. Simply moving from MERT to cutting-plane MIRA for parameter optimization yields improvement in most cases, corroborating existing work (Eidelman, 2012). Using incremental post-editing data to update translation grammars (G) yields further improvement in all cases evaluated. Gains are significantly larger for TED talks where translator feedback can bridge the gap between domains. Table 5 shows the aggregate percentages of rules in online grammars that are entirely new (extracted from post-editing instances only) or post-edit supported (superset of new rules). While percentages vary by data set, the overall trend is a combination of *learning* new vocabulary and reordering and *disambiguating* existing translation choices.

The introduction of a trigram Bayesian language model (L) yields mixed results: in some

[3] `http://www.ted.com/talks`

[4] Derivation features consist of word count, discretized rule-level non-terminal count (0, 1, or 2), glue rule count, and out-of-vocabulary pass-through count.

Base MERT	and changing the definition of what the **Zona Cero** is .
G+L+M	and the changing definition of what the **Ground Zero** is .
Reference	and the changing definition of what **Ground Zero** is .
Base MERT	was that when we **side by side** comparisons with **coal** , **timber**
G+L+M	was that when we did **side-by-side** comparisons with **wood charcoal** ,
Reference	was when we did **side-by-side** comparisons with **wood charcoal** ,
Base MERT	There was a way – **there was one** –
G+L+M	There was a way – **there had to be a way** –
Reference	There was a way – **there had to be a way** –

Table 4: Translation examples from baseline and fully adaptive systems of Spanish TED talks into English. Examples illustrate (from top to bottom) learning translations for new vocabulary items, selecting correct translation candidates for the domain, and learning domain-appropriate phrasing.

cases it leads to slight improvement and in others, degradation. It appears that a static but large 4-gram language model often outperforms an incrementally updated but smaller trigram model. Further, learning a single weight for the Bayesian model can lead to a harmful mismatch. As a tuning pass over the development corpus proceeds, the model incorporates additional data and MIRA learns a weight corresponding to its predictive ability at the end of the corpus. During decoding, *all* sentences are translated with this language model weight, even before the model can adequately adapt itself to the target domain. This problem is alleviated in our fully adaptive system.

Using cutting-plane MIRA to incrementally update weights during decoding (M) also leads to mixed results, frequently resulting in both small increases and decreases in score. This could be due to the noise incurred when making small adjustments to static features after each sentence: depending on the similarity between the previous and current sentence and the limit of the step size (regularization strength), a parameter update may slightly improve or degrade translation.

Finally, we see significantly larger gains for our fully adaptive system (G+L+M) that combines adaptive translation grammars and language models with online parameter updates. In many cases, the difference between the baseline systems and our adaptive system is *greater* than the sum of the differences from our individual techniques, demonstrating the effectiveness of combining online learning methods. Our final system has two key advantages over any individual extension. First, incremental updates from MIRA can *rescale* weights for features that *change* over time, keeping the model consistent. Second, the Bayesian language model's out-of-vocabulary feature can *discriminate* between true OOV items and vocabulary items in the post-editing data not present in the monolingual data. By contrast, the only OOVs in the baseline system are untranslated items, as the target side of the bitext is included in the language model training data. This interplay between the adaptive components in our translation system leads to significant gains over MERT and MIRA baselines. Table 4 contains examples from our system's output that exemplify key improvements in translation quality. With respect to performance, our fully adaptive system translates an average of 1.5 sentences per second per CPU core. The additional cost incurred updating translation grammars and language models is less than one second per sentence (though the baseline cost of on-demand grammar extraction can be up to a few seconds). In total, the system is well within the acceptable speed range needed to function in real-time human translation scenarios.

6.4 Evaluation Using Post-Edited References

The 2012 ACL Workshop on Machine Translation (Callison-Burch et al., 2012) makes available a set of 1832 English–Spanish parallel news source sentences, independent references, initial MT outputs, and post-edited MT outputs. The employed MT system is trained on largely the same resources as our own English–Spanish system, granting the opportunity for a much closer approximation to an actual post-editing task; our system configurations score between 54 and 56 BLEU against the sample MT, indicating that humans post-edited translations similar but not identical to our own. We split the data into development and test sets, each 916 sentences, and run 3 iterations of optimizing on the development set and evaluating on the test set with both the MERT baseline and our G+L+M system on both types of references. Using independent references for tuning and evaluation (as before), our system yields an improvement of 0.6 BLEU (23.3 to 23.9). With post-edited references, our system yields an improvement of 1.3 BLEU (43.0 to 44.3). This provides strong evidence that our adaptive systems would provide better translations (both in terms of absolute quality and improvement over a standard baseline) for real-world post-editing scenarios.

7 Related Work

Prior work has led to the extension of standard phrase-based translation systems to make use of incrementally available data.[5] Approaches generally fall into categories of adding new data to translation models and of using incremental data to adjust model parameters (feature weights). In the first case, Nepveu et al. (2004) use cache-based translation and language models to incorporate data from the current document into a computer-aided translation scenario. Ortiz-Martínez et al. (2010) augment a standard translation model by storing sufficient statistics in addition to feature scores for phrase pairs, allowing feature values to be incrementally updated as new sentence pairs are available for phrase extraction. Hardt and Elming (2010) demonstrate the benefit of maintaining a distinction between background and post-editing data in an adaptive model with simulated post-editing. Though not targeted at post-editing applications, the most similar work to our online grammar adaptation is the stream-based translation model described by Levenberg et al. (2010). The authors introduce a dynamic suffix array that can incorporate new training text as it becomes available. Sanchis-Trilles (2012) proposes a strategy for online language model adaptation wherein several smaller domain-specific models are built and their scores interpolated for each sentence translated based on the target domain.

Focusing on incrementally updating model parameters with post-editing data, Martínez-Gómez et al. (2012) and López-Salcedo et al. (2012) show improvement under some conditions when using techniques including passive-aggressive algorithms, perceptron, and discriminative ridge regression to adapt feature weights for systems initially tuned using MERT. This work also uses reference translations to simulate post-editing. Saluja et al. (2012) introduce a support vector machine-based algorithm capable of learning from binary-labeled examples. This learning algorithm is used to incrementally adjust feature weights given user feedback on whether a translation is “good” or “bad”. As with our work, this strategy can be used during both optimization and decoding.

Finally, Simard and Foster (2013) apply a pipeline solution to the post-editing task wherein a second stage automatic post-editor (APE) system learns to replicate the corrections made to initial MT output by human translators. As incremental data accumulates, the APE (itself a statistical phrase-based system) attempts to “correct” the MT output before it is shown to humans.

8 Conclusion

Casting machine translation for post-editing as an online learning task, we have presented three methods for incremental model adaptation: adding data to the indexed bitext from which grammars are extracted, updating a Bayesian language model with incremental data, and using an online discriminative parameter update during decoding. These methods, which allow the system to handle all data in a uniform way, are applied to a strong baseline system optimized using MIRA in conjunction with simulated post-editing. In addition to showing gains for individual methods under various circumstances, we report super-additive improvement from combining our techniques to produce a fully adaptive system. Improvements generalize over language and data scenarios, with the greatest gains realized in blind out-of-domain tasks where the system must rely heavily on post-editor feedback to improve quality. Gains are also more significant when using offline post-edited references, showing promise for applying our techniques to real-world post-editing tasks. All software used for our online model adaptation experiments is freely available under an open source license as part of the `cdec` toolkit.[6]

Acknowledgements

This work is supported in part by the National Science Foundation under grant IIS-0915327, by the Qatar National Research Fund (a member of the Qatar Foundation) under grant NPRP 09-1140-1-177, and by the NSF-sponsored XSEDE program under grant TG-CCR110017.

[5] Prior to phrase-based systems, NISHIDA et al. (1988) use post-editing data to correct errors in transfer-based MT.

[6] `http://www.cs.cmu.edu/~mdenkows/cdec-realtime.html`

References

[Callison-Burch et al.2012] Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. Association for Computational Linguistics.

[Carl et al.2011] Michael Carl, Barbara Dragsted, Jakob Elming, Daniel Hardt, and Arnt Lykke Jakobsen. 2011. The process of post-editing: A pilot study. *Copenhagen Studies in Language*, 41:131–142.

[Cettolo et al.2012] Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the Sixteenth Annual Conference of the European Association for Machine Translation*.

[Chiang2007] David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33.

[Chiang2012] David Chiang. 2012. Hope and fear for discriminative training of statistical translation models. *Journal of Machine Learning Research*, pages 1159–1187, April.

[Clark et al.2011] Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA, June. Association for Computational Linguistics.

[Crammer et al.2006] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, pages 551–558, March.

[Dyer et al.2010] Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Ture, 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 *Proceedings of the ACL 2010 System Demonstrations*, pages 7–12, Uppsala, Sweden, July. Association for Computational Linguistics.

[Dyer et al.2013] Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[Eidelman2012] Vladimir Eidelman. 2012. Optimization strategies for online large-margin learning in machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 480–489, Montréal, Canada, June. Association for Computational Linguistics.

[Guerberof2009] Ana Guerberof. 2009. Productivity and quality in mt post-editing. In *Proceedings of MT Summit XII - Workshop: Beyond Translation Memories: New Tools for Translators MT*.

[Hardt and Elming2010] Daniel Hardt and Jakob Elming. 2010. Incremental re-training for post-editing smt. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas*.

[Heafield et al.2013] Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, August.

[Hopkins and May2011] Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

[Koehn2012] Philipp Koehn. 2012. Computer-aided translation. Machine Translation Marathon.

[Kuhn and de Mori1990] Roland Kuhn and Renato de Mori. 1990. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6).

[Levenberg et al.2010] Abby Levenberg, Chris Callison-Burch, and Miles Osborne. 2010. Stream-based translation models for statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 394–402, Los Angeles, California, June. Association for Computational Linguistics.

[Lopez2008a] Adam Lopez. 2008a. Machine translation by pattern matching. In *Dissertation, University of Maryland*, March.

[Lopez2008b] Adam Lopez. 2008b. Tera-scale translation models via pattern matching. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 505–512, Manchester, UK, August. Coling 2008 Organizing Committee.

[López-Salcedo et al.2012] Francisco-Javier López-Salcedo, Germán Sanchis-Trilles, and Francisco Casacuberta. 2012. Online learning of log-linear weights in interactive machine translation. *Advances in Speech and Language Technologies for Iberian Languages*, pages 277–286.

[Macherey et al.2008] Wolfgang Macherey, Franz Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 725–734, Honolulu, Hawaii, October. Association for Computational Linguistics.

[Manber and Myers1993] Udi Manber and Gene Myers. 1993. Suffix arrays: A new method for online string searches. *SIAM Journal of Computing*, 22:935–948.

[Martínez-Gómez et al.2012] Pascual Martínez-Gómez, Germán Sanchis-Trilles, and Francisco Casacuberta. 2012. Online adaptation strategies for statistical machine translation in post-editing scenarios. *Pattern Recognition*, 45:3193–3203.

[Nepveu et al.2004] Laurent Nepveu, Guy Lapalme, Philippe Langlais, and George Foster. 2004. Adaptive language and translation models for interactive machine translation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 190–197, Barcelona, Spain, July. Association for Computational Linguistics.

[NISHIDA et al.1988] Fujio NISHIDA, Shinobu TAKAMATSU, Tadaaki TANI, and Tsunehisa DOI. 1988. Feedback of correcting information in postediting to a machine translation system. In *Proc. of COLING*.

[Och2003] Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July. Association for Computational Linguistics.

[Ortiz-Martínez et al.2010] Daniel Ortiz-Martínez, Ismael García-Varea, and Francisco Casacuberta. 2010. Online learning for interactive statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 546–554, Los Angeles, California, June. Association for Computational Linguistics.

[Papineni et al.2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

[Parker et al.2011] Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English Gigaword Fifth Edition, June. Linguistic Data Consortium, LDC2011T07.

[Przybocki2012] Mark Przybocki. 2012. Nist open machine translation 2012 evaluation (openmt12). http://www.nist.gov/itl/iad/mig/openmt12.cfm.

[Saluja et al.2012] Avneesh Saluja, Ian Lane, and Ying Zhang. 2012. Machine translation with binary feedback: a large-margin approach. In *Proceedings of the Tenth Biennial Conference of the Association for Machine Translation in the Americas*.

[Sanchis-Trilles2012] Germán Sanchis-Trilles. 2012. Building task-oriented machine translation systems. In *Ph.D. Thesis, Universitat Politcnica de Valncia*.

[Simard and Foster2013] Michel Simard and George Foster. 2013. PEPr: Post-edit propagation using phrase-based statistical machine translation. In *Proceedings of the XIV Machine Translation Summit*, pages 191–198,, September.

[Tatsumi2009] Midori Tatsumi. 2009. Correlation between automatic evaluation metric scores, post-editing speed, and some other factors. In *Proceedings of the Twelfth Machine Translation Summit*.

[Teh2006] Yee Whye Teh. 2006. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proc. of ACL*.

[Zhao et al.2004] Bing Zhao, Matthias Eck, and Stephan Vogel. 2004. Language model adaptation for statistical machine translation with structured query models. In *Proc. of COLING*.

[Zhechev2012] Ventsislav Zhechev. 2012. Machine Translation Infrastructure and Post-editing Performance at Autodesk. In *AMTA 2012 Workshop on Post-Editing Technology and Practice (WPTP 2012)*, pages 87–96, San Diego, USA, October. Association for Machine Translation in the Americas (AMTA).