Machine Translation with Binary Feedback: a Large-Margin Approach

Avneesh Saluja
Carnegie Mellon University
avneesh@cmu.edu

Ian Lane
Carnegie Mellon University
ianlane@cs.cmu.edu

Ying Zhang
Carnegie Mellon University
joy@cs.cmu.edu

Abstract

Viewing machine translation as a structured classification problem has provided a gateway for a host of structured prediction techniques to enter the field. In particular, large-margin structured prediction methods for discriminative training of feature weights, such as the structured perceptron or MIRA, have started to match or exceed the performance of existing methods such as MERT. One issue with structured problems in general is the difficulty in obtaining fully structured labels, e.g., in machine translation, obtaining reference translations or parallel sentence corpora for arbitrary language pairs. Another issue, more specific to the translation domain, is the difficulty in online training of machine translation systems, since existing methods often require bilingual knowledge to correct translation output online. We propose a solution to these two problems, by demonstrating a way to incorporate binary-labeled feedback (i.e., feedback on whether a translation hypothesis is a "good" or understandable one or not), a form of supervision that can be easily integrated in an online manner, into a machine translation framework. Experimental results show marked improvement by incorporating binary feedback on unseen test data, with gains exceeding 5.5 BLEU points.

1 Introduction

Structured prediction is an umbrella term used for a variety of machine learning algorithms and frameworks. The common thread that links these approaches together is that when predicting over a particular variable, the relationships between that variable and others are taken into account (in the form of a "structure") . An example of a simple and effective structured prediction model is the hidden Markov model, wherein we leverage the chain structure of the variables when computing the most probable hidden state sequence. These models can also be generalized to tree-like structures, as in probabilistic context free grammars. Applications of structured prediction models in natural language processing, computer vision, bioinformatics, and other fields are numerous.

The coupling introduced in the input or output space complicates matters, in that the sizes of the input or output spaces ($\mathcal{X}$ and $\mathcal{Y}$ respectively) are exponential in the cardinality of these spaces. Past approaches have looked at ways to handle these exponential constraints, a popular method being to assume the loss functions decompose locally as per the structure of the model (since generally, smaller structures lead to easier inference and learning), and then present the globally consistent solution as a simple combination of the local solutions. In addition to the more difficult learning and inference problems, there are also obstacles in obtaining the complete labeling for structured data. For instance, in a simple part-of-speech (POS) tagging task, a fully-labeled structured prediction example needs POS tags for all of the words in a given sentence; partial labelings of the output are of limited use.

Structured prediction labelings are thus difficult to obtain, however obtaining some form of weak supervision regarding the examples is not as difficult. For the same POS example, instead of obtaining POS

tags for each word in that particular context, which may require significant human annotator effort, an alternative is to consider soliciting only binary labels for the example. In this situation, for every input string that needs to be tagged, we have a tagged hypothesis associated with that string, and a binary (positive or negative) label on the hypothesis which marks it as correct or incorrect. This alternative is motivated by practical considerations and active learning, i.e., a "human in the loop" approach: it is easier for an end user to mark a hypothesis generated by a system as "correct" or "incorrect" compared to the user having to annotate every word or phrase in some manner. The question now is, how does the system incorporate this type of feedback to improve performance over time?

In this work, we focus on the incorporation of binary-labeled feedback in statistical machine translation (SMT). The proposed approach addresses the lack of adaptation in current translation systems where the models remain static after they are trained. If a system produces a poor translation, this information is important and can potentially improve future translations. Users may feel especially frustrated when the system repeats a type of error on a consistent basis. And from a learning perspective, user feedback provides new information in addition to the original monolingual and bilingual training schemes.

In particular, we focus on large-margin discriminative training for SMT, and augment a latent structural SVM-based algorithm for learning feature weights to also learn over the binary-labeled examples. After proposing a method to incorporate weakly labeled feedback, we evaluate our approach on a real-world translation task, with significant improvements (in excess of 5.5 BLEU points) over approaches that do not incorporate such feedback, and equally convincing improvements over MERT (more than 4 BLEU points), currently a popular parameter tuning algorithm for SMT. Our approach can be seen both as a way to incorporate other forms of supervision in machine translation (in addition to full reference translations and reference translation-derived costs or scores), and on the practical side as a simple, efficient method to incorporate user feedback in an effort to move towards an effective solution for online training of translation systems.

2 Large-Margin Training with Binary Feedback for MT

In machine translation, the objective is to map an input string $\mathbf{x}$ in one language (the "source") to a string $\mathbf{y}$ in another language (the "target"). The primary basis for translation currently is statistical, and can be easily expressed as a linear structured prediction problem (Liang et al., 2006):

$$f(\mathbf{x}; \mathbf{w}) = \arg\max_{\mathbf{y},\mathbf{h}} \mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y}, \mathbf{h}) \qquad (1)$$

where $\mathbf{x}$ is the source sentence, $\mathbf{y}$ is the target sentence, and $\mathbf{h}$ is a set of variables corresponding to the hidden structure that is used to map from the source to the target. In the POS tagging example, the relationship between $\mathbf{x}$ and $\mathbf{y}$ is observed, but in tasks such as translation, the relationship is unknown from the data given to us (parallel sentence pairs), and is often referred to as the derivation. In phrase-based MT models (Koehn et al., 2003), $\mathbf{h}$ corresponds to phrase segmentations of source and target sentences, and a bijection between source and target phrases. $\Phi(\cdot)$ is a feature vector defined over the source, target, and hidden derivation when going from source to target. Common components of this vector may be the log probability of the target string from the language model, or the score of translating from source to target based on a phrase model. Lastly, $\mathbf{w}$ is the weight or parameter vector; it is this set of parameters that we need to learn in the learning stage of our method.

2.1 Large-Margin Training in MT

There are a number of formulations that can be used to learn the vector $\mathbf{w}$. A common approach in current SMT work is the Minimum Error Rate Training (MERT) scheme of (Och, 2003), wherein the goal is to minimize a task-specific loss, mostly BLEU (Papineni et al., 2002) or some other evaluation metric. The search procedure is not optimal and leads to local minima, but in practice the results seem to be acceptable, due to the direct error or cost minimization procedure.

Recent work has attempted to incorporate the notion of margin maximization when learning the parameter vector $\mathbf{w}$. The basic idea is to incorporate the margin in the constraints, in that we require the

margin between the "true" label (i.e., the reference translation in our scenario) and the proposed label (i.e., the hypothesis translation) to be proportional to the "cost" of selecting the hypothesis over the reference. The cost is where we incorporate an external metric, like BLEU, with the aim being to update the weights such that we score the reference higher than our hypothesis by an amount proportional to the margin. In effect, the algorithm chooses the "most dangerous competitor" as a negative example, namely an example that has high model score, as per Equation 1, but also has a high cost, and updates away from this example and towards the reference. We use the term "cost-augmented hypothesis" to refer to this negative example, since we take into account the cost when selecting this hypothesis[1].

Regarding the choice of the reference, as in Liang *et al.* (2006), we note that it may not always be appropriate to use the actual reference translation ("bold" updating), as the reference may not always be reachable by the model, so we take the highest scoring translation that is actually achievable by our model ("local" updating). In practice, this choice amounts to picking the hypothesis in the k-best list with the highest sentence-level BLEU, since we continue with the standard MT parameter tuning assumption of restricting the search space to a series of k-best lists. Additional details on other approaches that incorporate the maximum margin idea are presented in Section 4.

In this work, we adopt the latent structural SVM of (Yu and Joachims, 2009), as the framework allows us to handle hidden variables in a structured, maximum-margin setting. The objective function is:

$$\begin{aligned}
&\min_{\mathbf{w}} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \\
&C_1 \sum_{i=1}^{n} \left(\max_{\mathbf{y},\mathbf{h}} [\mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y}, \mathbf{h}) + \Delta(\mathbf{y}_i, \mathbf{y})] \right) \\
&- C_1 \sum_{i=1}^{n} \left(\max_{\mathbf{h}} \mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y}_i, \mathbf{h}) \right) \qquad (2) \\
&= \min_{\mathbf{w}} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \\
&C_1 \sum_{i \in S} \left(\max_{\mathbf{y}} [\Delta(\mathbf{y}_i, \mathbf{y}) - \mathbf{w} \cdot \Phi_{\mathbf{y}_i,\mathbf{y}}(\mathbf{x})] \right) \qquad (3)
\end{aligned}$$

[1]it is also called the "loss-augmented hypothesis" in the literature, but we choose to avoid this term lest it be confused with the loss function instead of the extrinsic cost function.

where in Equation 2, $\mathbf{y}_i$ is the reference translation, henceforth referred to as the oracle, $\mathbf{y} \in \mathcal{Y}$ is the hypothesis, $\Delta(\mathbf{y}_i, \mathbf{y})$ is the extrinsic cost function that provides a measure of how much worse it is to choose the hypothesis instead of the reference (we use difference in BLEU between the oracle and the hypothesis), n is the number of examples (sentences) in our training corpus, λ is the regularization strength, and C_1 is a constant (it can be interpreted as the step size in the optimization, as discussed in Section 2.3). We also write this objective in a more concise form in Equation 3, where the set S refers to the training set that contains reference translations for source-side sentences, and $\Phi_{\mathbf{y}_i,\mathbf{y}}(\mathbf{x}) = \Phi(\mathbf{x}, \mathbf{y}_i, \mathbf{h}_i) - \Phi(\mathbf{x}, \mathbf{y}, \mathbf{h}^*)$ such that $\mathbf{h}_i = \arg\max_{\mathbf{h} \in \mathcal{H}} \mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y}_i, \mathbf{h})$ and $\mathbf{h}^* = \arg\max_{\mathbf{h} \in \mathcal{H}} \mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y}, \mathbf{h})$. We are aware of only one other work in translation that uses the structural SVM for training (Cherry and Foster, 2012), although our approach was developed independently and thus has several key differences, primarily a different optimization approach (see Section 2.3 for more details).

In practice, what the above formulation amounts to is that for every sentence, we choose the highest-scoring hypothesis (in terms of an external metric like BLEU) amongst the hypotheses in a k-best list to be our oracle $\mathbf{y}_i$, and the hypothesis that maximizes a combination of the model score and an external cost as our cost-augmented hypothesis, and update our weights towards the former and away from the latter. Note that when selecting the cost-augmented hypothesis, we weight the model score and the external cost function equally (and also convert the BLEU score into log space so that the quantities are on the same scale); we plan to investigate different scalings in future work. Similarly, while we only use an external cost to select the oracle, one can also incorporate the model score in that selection too.

2.2 Incorporating Binary Feedback

The objective function in Equation 2 requires that we have a structured label, $\mathbf{y}_i$, for every training example we see. In the case of MT, we need a reference translation for every sentence, i.e., we need a sentence-aligned parallel corpus. We note that even though we may not always update towards the ref-

erence translation (as it may not even be achievable by our model), we need the reference translation to select our oracle, the achievable translation with the highest BLEU. Obtaining structured labels, or reference translations is not a trivial or easy task, especially in the online setting. If a user receives an incorrect hypothesis from the SMT system, it would either be cumbersome to provide the reference, or it may not even be possible if the user is monolingual with respect to the language pair, a fairly typical end-user for machine translation. What is possible and easier however, is for the user to provide a quick, binary "good" or "bad" response to the translation. How do we incorporate this form of weak supervision into our objective function?

In response to this question, we propose to augment the concise objective function in Equation 3 in a similar manner to (Chang et al., 2010):

$$\min_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + C_1 \sum_{i \in S} \left(\max_{\mathbf{y}} [\Delta(\mathbf{y}_i, \mathbf{y}) - \mathbf{w} \cdot \Phi_{\mathbf{y}_i,\mathbf{y}}(\mathbf{x})] \right) + C_2 \sum_{i \in B} \max \left(0, 1 - \ell_i \max_{\mathbf{h}} \mathbf{w} \cdot \Phi_B(\mathbf{x}, \hat{\mathbf{y}}_i, \mathbf{h}) \right) \quad (4)$$

where C_2 is a constant, B is the set of binary-labeled ("good" and "bad") sentences that form the positive and negative corpora, $\ell_i = 1$ if $i \in B^+$ (the positive examples), $\ell_i = -1$ if $i \in B^-$ (the negative examples), $\hat{\mathbf{y}}_i$ is the target-side hypothesis with corresponding label ℓ_i, and $\Phi_B(\mathbf{x}, \hat{\mathbf{y}}_i, \mathbf{h}) = \frac{\Phi(\mathbf{x}, \hat{\mathbf{y}}_i, \mathbf{h})}{\kappa(\mathbf{x})}$, where $\kappa(\mathbf{x})$ is a scaling factor that equal the length of the source sentence $\mathbf{x}$, as the features are sensitive to the length of the source sentence. Note that we use the hinge loss in the augmented term that handles binary-labeled examples, and further note that our binary feedback-related term includes a maximization over $\mathbf{h}$, the set of all derivations that yield the generated target-side hypothesis $\hat{\mathbf{y}}_i$ with binary label ℓ_i. Recall that the second term in Equation 4 already contains the maximizations over hidden derivations $\mathbf{h}$, as per Equation 3.

As per our knowledge, this work is the first to use the augmented form of Equation 4 in machine translation; previous applications have focused on arguably simpler problems.

2.3 Optimization for Learning

There are several options one could pursue when it comes to optimizing the objective function in Equation 4. (Chang et al., 2010) use an iterative procedure similar to CCCP (concave convex procedure), with a cutting planes strategy as per (Yu and Joachims, 2009). Given the online nature of our problem of updating parameter weights when provided with positive or negative feedback on a particular hypothesis, we feel it is more appropriate to use a structured subgradient-based method, as described by (Ratliff et al., 2007). This optimization method is conceptually simpler than the cutting planes and CCCP-based optimization, and also leads to fast rates of convergence. Implementation of the algorithm only requires the computation of the gradient of Equation 4. Let $F(\mathbf{w}_t)$ represent Equation 4, then the update step is:

$$\begin{aligned}
\mathbf{w}_{t+1} &= \mathbf{w}_t - \gamma \nabla F(\mathbf{w}_t) \\
&= \mathbf{w}_t - \gamma \left(\lambda \mathbf{w}_t + C_1 \sum_{i \in S} -\Phi_{\mathbf{y}_i,\mathbf{y}}(\mathbf{x}) + C_2 \sum_{i \in B} -\ell_i \max_{\mathbf{h}} \Phi_B(\mathbf{x}, \hat{\mathbf{y}}_i, \mathbf{h}) \right) \\
&= (1 - \gamma\lambda)\mathbf{w}_t + \gamma \left(C_1 \sum_{i \in S} \Phi(\mathbf{x}, \mathbf{y}_i, \mathbf{h}_i) - \Phi(\mathbf{x}, \mathbf{y}, \mathbf{h}) + C_2 \sum_{i \in B} \ell_i \max_{\mathbf{h}} \Phi_B(\mathbf{x}, \hat{\mathbf{y}}_i, \mathbf{h}) \right) \quad (5)
\end{aligned}$$

where γ is the gradient descent step size, which is expressed as a parameter independent of the iteration t, i.e., a fixed step size. The MIRA approach (Watanabe et al., 2007; Chiang et al., 2008) naturally provides adaptive step sizes, and extending our binary-labeled framework to this regime remains future work.

The intuition of Equation 5 is apparent: in the first term we have the regularization effect, in the second and third terms we essentially add to the parameter weights an amount proportional to the feature values of the oracle translation, and subtract an amount proportional to the feature values of the hypothesis translation (i.e., the "most dangerous competitor"). The last term is the one that deals with the binary-labeled feedback, and essentially states that if the feedback is positive, then we add to the parameters an amount proportional to the feature values of that particular hypothesis translation, otherwise we subtract.

We can either update the weights after every sen-

tence i (the online or stochastic version) or at periodic intervals (the batch version), chosen to be after every pass through a binary-labeled sub-corpus. Updating the weights technically results in a different k-best list for each sentence, or in other words a moving oracle for the fully supervised corpus, whereas our analysis hinges upon the k-best lists remaining fixed (Gimpel and Smith, 2012). In practice, we reselect the oracle in a manner similar to (Cherry and Foster, 2012). In our experiments (Section 3), we found that a batch update seemed to provide a smoother improvement over iterations, and thus we chose this updating scheme for our final results.

3 Evaluation

The aim of the evaluation was to show that our method effectively incorporates weak supervision in the form of binary labels to improve BLEU over time. We evaluated our proposed approach on a Chinese-English dataset from the travel domain, the BTEC (Basic Travel Expressions Corpus) dataset from the IWSLT 2009 evaluation campaign (Paul, 2009). The corpus statistics are provided in Table 1.

Corpus	Sentences	Words (Source)
Training	182,288	1,293,520
Development	507	3420
Test	246	1543

Table 1: Number of words and sentences for the training, tuning, and test corpora that we used

We used a hierarchical phrase translation (synchronous context-free grammar) framework for our experiments. We used Thrax[2] for the grammar extraction, GIZA++ (Och and Ney, 2003) for the word-level alignments (using the `grow-diag-final-and` heuristic), SRILM (Stolcke, 2002) and KenLM (Heafield, 2011) for extracting and building the LM binaries respectively, and cdec (Dyer et al., 2010) for decoding. We also used cdec's libraries to implement our algorithm, and are in the process of merging our code with the mainstream package.

[2]http://cs.jhu.edu/ jonny/thrax/

3.1 Experimental Setup

Our algorithm relies on some form of feedback to provide the weak supervision of binary labels. While the motivation for our approach is based on a "human in the loop" framework for providing these labels, for the purposes of our experiments we resorted to synthetically generating this feedback. The experimental flow is shown in Figure 1.

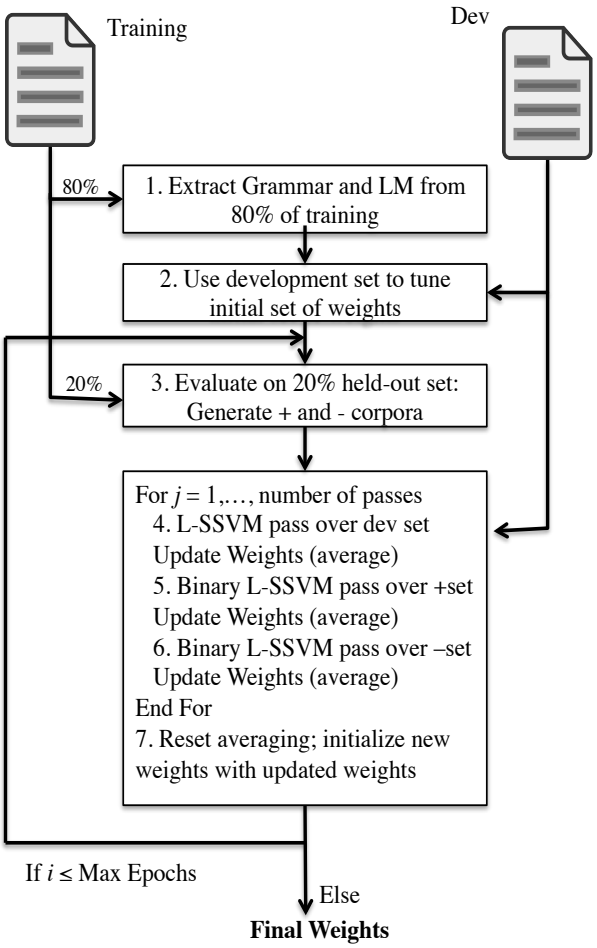

Figure 1: Flowchart for the experimental setup as described in Section 3.1.

First, we took 20% of our training set for the purposes of feedback generation. With the remaining 80% of the data, we extracted a grammar and filtered it (Figure 1, Step 1) so that for any source right-hand side, we kept at most 15 distinct target right-hand side rules, sorted by the phrase target conditional probability given the source, i.e., $P(e|f)$. The resulting grammar size consisted of 1,133,469 rules. We also extracted a 3-gram language model from

the remaining 80% of the data, and this was the only language model we used during decoding.

We then bootstrapped our system using the latent structural SVM algorithm without any binary feedback (Equation 3) and the development set to get an initial baseline system (Figure 1, Step 2). This initial system was generated by updating the weights over 10 passes through the development set, where we cumulatively averaged the weights between each pass. Using the bootstrapped set of parameter values, we evaluated on our held-out 20% of the training data (Figure 1, Step 3), and using the reference translations for this held-out set, evaluated BLEU at the sentence-level[3]. Post-evaluation, translations that achieved a sentence-level BLEU greater than 80 (on a scale of 0 to 100) were deemed to have received "positive" feedback, and sentences that achieved a sentence-level BLEU less than 20 were deemed to have received "negative" feedback.

After having obtained a set of positive and negative sentences, we proceeded with our proposed algorithm, optimizing the objective function in Equation 4 using batch structured gradient descent (Section 2.3). Within each epoch, we go through the development (Figure 1, Step 4), positive (Step 5), and negative (Step 6) corpora 10 times (corresponding to the 10 passes), cumulatively averaging the weights over passes. After updating the weights, we evaluated the 20% held-out set again (Figure 1, Step 3), and continued this methodology for a total of 10 epochs. The averaging of weights was reset as we moved between epochs.

We used the standard set of small-scale features for our experiments, namely a language model and language model OOV feature, 6 phrase model or grammar features (the lexical and phrase conditional probabilities in both directions, as well as the phrase penalty and glue rule features), a pass-through and a word penalty feature. For future work, we plan to extend our approach to numerous sparse, overlapping features.

As a basis for comparison, we also tuned the parameter weights using MERT, firstly with 80% of the training data (i.e., without the held-out portion), tuned on the same development set, and evaluated on the same test set, and secondly with 100% of the training data. Results were averaged over 5 MERT runs for each setup. The language model and grammar used were the same as those used in the latent SSVM implementation.

[3]we used the NIST `mteval-v13.pl` script with the sentence-level BLEU flag.

3.2 Results

Results for the Chinese-English BTEC dataset over 10 epochs are presented in Figure 2, where we show the learning curves as the number of iterations proceeds for the held-out "feedback generation" set, as well as the test set and the development set. In addition, the dashed horizontal lines provide evaluation results based on the two MERT baselines (80% and 100% of training data). Given that the "feedback generation" set is used to determine positive and negative sentences, the steady improvement in BLEU (i.e., the number of positive sentences being generated vs. the number of negative sentences) on this set is in line with what we would expect, and the final improvement amounts to 3.18 BLEU. The appealing aspect of our approach is the significant and smooth improvement on the unseen test set, from a baseline BLEU score of 37.97, to an eventual score of 43.15 after 10 epochs, and achieving a highest score of 43.64 (after epoch 9). The maximum improvement is 5.67 BLEU points. Our results compare favorably with the MERT baselines - even with 100% of the training data, the MERT-tuned weights result in an average BLEU score of 38.94, more than 4 BLEU points away from the eventual binary feedback-based result.

We also decided to investigate an alternative experimental setup, whereby the "positive" hypotheses are assumed to be good enough to be reference translations, the development set gets augmented with these translations and a subsequent re-training is done on the augmented development set (for 10 epochs). The results are presented in Figure 3, and show that while the performance on the 20% held-out set improves (by a BLEU score of 4.59, higher than our proposed approach), the generalization to the unseen test set is significantly impacted: the best improvement in terms of BLEU on the test set is 1.75 points for this method. This finding underscores the importance of negative examples especially in training a translation system. In addition, the augmentation constantly increases the size of the development

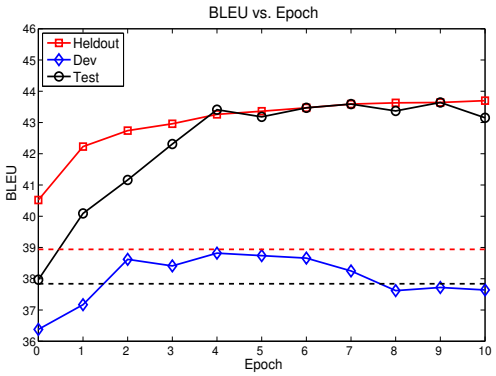

Figure 2: BLEU vs. Epoch for the iterative weakly supervised experiments. One sees a significant increase in BLEU by incorporating binary labeled data. The dashed lines represent the MERT baselines evaluated on the test set only, red for 100% of the training data, and black for 80% of the training.

set at each epoch. In fact, while our development set started off with 507 sentences (Table 1), after 10 epochs we had almost 90,000 sentences in the development set, which significantly increases the time taken for parameter tuning.

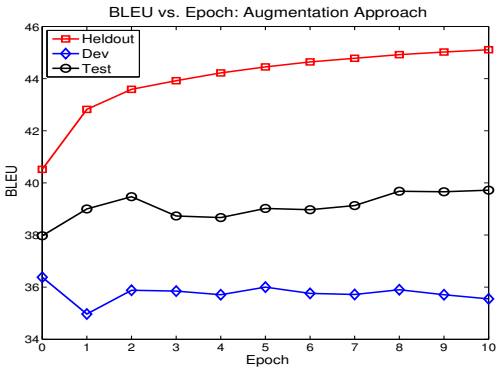

Figure 3: BLEU vs. Epoch for the experiments where we augment the development set with sentences from the heldout set with a BLEU score greater than a threshold (the "positive" feedback data). Performance on the heldout set is strong, as one can expect, but generalization is relatively poor.

There are several additional parameters introduced by the algorithm, as presented in Equation 5: the regularization strength λ and the supervised (C_1) and positive/negative (C_2) step sizes. We set $\lambda = 0.001$, and $C_1 = C_2 = 1 \times \mathrm{e}^{-6}$, after experimenting with several different values.

4 Related Work

Large-margin training started to become popular in natural language processing after the structured perceptron of (Collins, 2002). The first application of large-margin training in MT (at the training and decoding level, as opposed to restricted to k-best list reranking) was by (Liang et al., 2006), where they applied the structured perceptron to train the weights of a high number of sparse features. Subsequent advancements in large margin training such as the application of MIRA (Crammer and Singer, 2003) include (Watanabe et al., 2007), (Blunsom et al., 2008), and (Chiang et al., 2008).

The latent structural SVM formulation was presented by (Yu and Joachims, 2009), to handle hidden or latent structures as is prevalent in natural language processing, an extension to the "slack rescaling" (Tsochantaridis et al., 2005) and "margin rescaling" (Taskar et al., 2003) versions of the structural SVM. The structured subgradient method for optimization was proposed by (Ratliff et al., 2007), and (Chang et al., 2010) presented the addition of loss functions for binary-labeled data. We adapted the algorithm and made several MT-specific modifications, as well as the presentation of this method in a semi-online manner.

Some work has attempted to incorporate user feedback in SMT. (Ortiz-Martinez et al., 2010) introduce an online learning approach for an interactive SMT system, where the sufficient statistics of the generative models are updated incrementally. Their system is presented as an assistive tool for bilingual human translators. A monolingual speaker would find it difficult to provide the level of feedback required by the system without prior knowledge of the meaning of the sentence to be translated. Even for a bilinguals, the effort required in correcting hypotheses is excessive.

We note that our approach is somewhat similar to that proposed by (Hall et al., 2011), wherein the authors propose an augmented loss framework within a discriminative setting to handle additional datasets with arbitrary loss functions. They restrict their work to the structured perceptron optimized in an online fashion, and focus primarily on dependency parsing (albeit within a machine translation pipeline), whereas we look at improving transla-

tion quality in an end-to-end framework. These approaches would fall broadly into the "weak supervision" category of discriminative training, where we attempt to incorporate additional weaker or less informative sources of information to improve overall performance in the system.

5 Conclusion & Future Work

In this work, we presented a large-margin based approach to incorporate binary forms of feedback in an end-to-end machine translation framework. By adopting a suitable loss function over these binary-labeled examples, and performing the resulting optimization with structured gradient descent, we are able to achieve gains in excess of 5.5 BLEU points with the additional binary-labeled feedback.

In the future, we would like to work on numerous (i.e., more than a million) sparse, overlapping features, as well as test on additional language pairs.

Acknowledgments

We sincerely thank the anonymous reviewers for their insightful comments and suggestions for improvement. The latent SSVM algorithm (without binary feedback) was implemented by the first author as a class project in conjunction with Jeff Flanigan.

This work is partly supported by the Defense Advanced Research Projects Agency (DARPA) Transformative App program under the contract D11PC20022 and by the DARPA Broad Operational Language Translation (BOLT) project under Contract No. HR0011-12-C-0017.

References

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. A Discriminative Latent Variable Model for Statistical Machine Translation. In *Proceedings of the 46th Annual Meeting of the ACL-HLT*, pages 200–208, June.

Ming-Wei Chang, Vivek Srikumar, Dan Goldwasser, and Dan Roth. 2010. Structured output learning with indirect supervision. In *Proceedings of the 27th Annual International Conference on Machine Learning*, ICML '10.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *HLT-NAACL*, pages 427–436.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 224–233, Stroudsburg, PA, USA.

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 1–8.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, 3:951–991, March.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Johnathan 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 ACL*.

Kevin Gimpel and Noah A. Smith. 2012. Structured ramp loss minimization for machine translation. In *HLT-NAACL*, pages 221–231.

Keith Hall, Ryan T. McDonald, Jason Katz-Brown, and Michael Ringgaard. 2011. Training dependency parsers by jointly optimizing multiple objectives. In *EMNLP*, pages 1489–1499.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 761–768, Stroudsburg, PA, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, March.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of*

the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

D. Ortiz-Martinez, I. Garcia-Varea, and Francisco Casacuberta. 2010. Online Learning for Interactive Statistical Machine Translation. *HLT: NAACL 2010*, pages 546–554, June.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michael Paul. 2009. Overview of the iwslt 2009 evaluation campaign. In *Proceedings of IWSLT 2009*.

Nathan D. Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich. 2007. (online) subgradient methods for structured prediction. In *Artificial Intelligence and Statistics 2010*, AIStats '07.

Andreas Stolcke. 2002. SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference in Spoken Language Processing*.

Ben Taskar, Carlos Guestrin, and Daphne Koller. 2003. Max-margin markov networks. In *NIPS 2003*. MIT Press.

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. 2005. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6:1453–1484, September.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 764–773, Prague, Czech Republic, June. Association for Computational Linguistics.

Chun-Nam John Yu and Thorsten Joachims. 2009. Learning structural svms with latent variables. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1169–1176, New York, NY, USA. ACM.