Online Learning Methods For Discriminative Training of Phrase Based Statistical Machine Translation

Abhishek Arun and Philipp Koehn

School of Informatics
University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW, Scotland, UK
a.arun@sms.ed.ac.uk
pkoehn@inf.ed.ac.uk

Abstract

This paper investigates the task of training discriminatively a phrase based SMT system with millions of features using the structured perceptron and the Margin Infused Relax Algorithm (MIRA), two popular online learning algorithms. We also compare two different update strategies, one where we update towards an *oracle* translation candidate extracted from an *N*-best list vs a more aggressive approach in which we update towards an oracle extracted prior to training using a *minloss* decoder. We evaluate our different training algorithms on the Czech-English translation task. Our results show that while both learning algorithms achieve similar results, with the perceptron converging more rapidly, the aggressive update strategy performs significantly worse than the more conservative strategy corroborating Liang et al. (2006)'s findings.

1. Introduction

The direct maximum entropy model proposed by Och and Ney (2001) described a generalisation of the generative noisy-channel model of Brown et al. (1993) allowing the incorporation of additional knowledge sources in form of features. The standard way of training such a discriminative model in the community is to use minimum error rate training (MERT) (Och, 2003). As it name implies, such a training scheme tries to directly minimise the error on training data where the 'error' is evaluated using a loss function of one's choice (usually BLEU).

A major shortcoming of MERT is that it can only be used to train a model with a small number of features. Typically, state of the art phrase-based SMT systems use around a dozen features. Recent work has tried to address this problem, with both Liang et al. (2006) and Tillmann and Zhang (2006) presenting online perceptron/perceptron-like training schemes to learn parameters for models with millions of features.

Typically, discriminative training algorithms come in two flavours, firstly likelihood-based methods which require feature expectations and secondly, margin-based methods which require either an *N*-best list of best outputs or a marginal distribution across the graphical structure. The perceptron algorithm (Collins, 2002) is a much simpler alternative as it only requires an arg max computation, which is precisely what the decoder is set out to do.

While Liang et al. (2006) employ the standard perceptron algorithm in their work, Tillmann and Zhang (2006) present a variant in which the perceptron update is weighted with a factor dependent on the difference in translation score between the reference and the model guess and the difference in their loss. In this paper, we compare the perceptron algorithm to the Margin Infused Relaxed Algorithm(MIRA) (Crammer and Singer, 2003), a large-margin online algorithm that has given state-of-the-art results in other structured prediction tasks in NLP such as dependency parsing (McDonald et al., 2005).

Discriminative learning usually requires access to the gold standard, e.g. in discriminative dependency parsing (McDonald et al., 2005), the learning algorithm updates model parameters towards gold parse features. In the case of phrase based SMT, this is a problematic issue. Recall that phrases are extracted through heuristics from word alignments after symmetrisation (Koehn et al., 2003). Only phrases that are *consistent* with the alignments are extracted, therefore phrase-pairs that are required to reproduce the reference might end up **not being extracted**. Also, since multiple phrases are extracted from a given sentence-pair, in cases where the reference can be reproduced, there might be **multiple reference phrasal alignments**.

In this paper, we address this issue by comparing two strategies for selecting a gold standard translation example. The first strategy is the local updating of Liang et al. (2006), whereby the perceptron update is done towards the "best" candidate from an n-best list. The second strategy is the one employed by Tillmann and Zhang (2006) whereby the decoder is modified to use BLEU as its scoring function and is made to find the "best" candidate given the existing phrase table and source sentence. This is done as a pre-processing step and the extracted *surrogate* references are cached to be employed in the learning phase.

Our results show that the local updating strategy gives better results confirming Liang et al. (2006) findings that conservative updates are more effective.

2. The model

In phrase-based SMT models, the input ("foreign") sentence is segmented into so-called phrases, which are sequences of adjacent words that are not necessarily linguistically motivated. Each foreign phrase is mapped into the target language ("English"). Phrases are allowed to be reordered during translation; see Figure 1 for an illustration.

Similar to Liang et al. (2006), we model translation as a structured prediction task:

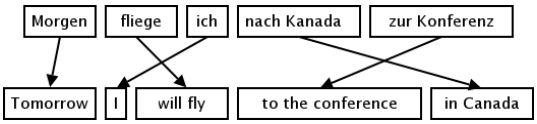

Figure 1: Phrase-Based SMT: Input sentence is segmented into phrases, which are then mapped onto output phrases.

$$\begin{aligned} s(\mathbf{x},\mathbf{y}) &= \sum_{(b_i,b_j,o)\in\mathbf{y}} s(b_i,b_j,o) \\ &= \sum_{(b_i,b_j,o)\in\mathbf{y}} \mathbf{w}\cdot\mathbf{f}(b_i,b_j,o) \end{aligned}$$

where $\mathbf{x}$ is an input (foreign) sentence, $\mathbf{y}$ is an output (English) sentence, $\mathbf{f}(b_i,b_j,o)$ is a multidimensional feature vector representation of sequence that produces phrase-pair b_j following phrase-pair b_i with orientation o and $\mathbf{w}$ the corresponding weight vector.
For example, a feature f_{101} in f could be:

$$f_{101}(b_i,b_j,o) = \begin{cases} 1 & \text{if } src(b_i) = \text{“fliege”} \\ & \wedge tgt(b_i) = \text{will fly} \\ 0 & \text{otherwise} \end{cases}$$

where $src(b_i)$ is the source phrase of the phrase pair b_i and $tgt(b_i)$ the target phrase of the phrase pair b_i.
Decoding in this model amounts to finding the $\mathbf{y}$ for a given $\mathbf{x}$ that maximises $s(\mathbf{x},\mathbf{y})$:

$$y' = \arg\max_y s(\mathbf{x},\mathbf{y})$$

Since exact decoding is intractable in SMT, we approximate the argmax using beam search.

3. Parameter estimation

In this section, we describe two online learning algorithms for learning the weight vector $\mathbf{w}$. As usual for supervised learning, we assume a training set $T = \{(x_t, y_t)\}_{t=1}^{T}$

3.1. Perceptron

The perceptron algorithm (Collins, 2002) is an incredibly simple algorithm that only requires an *argmax* computation.

$$w^{i+1} = w^i + \Theta(x_i, y_t) - \Theta(x_i, y_g) \qquad (1)$$

The perceptron algorithm consists in iterating several times over the training data, decoding each training instance one at a time. Each time the decoder's guessed solution is not equal to the correct solution, the weight vector is updated with the difference in feature vectors between the correct solution and the guess. The intermediate weight vectors are cumulated after each update and at test time, their averaged vector is used. This has been shown to alleviate overfitting (Collins, 2002).

3.2. MIRA

The Margin Infused Relaxed Algorithm (MIRA) (Crammer and Singer, 2003) is a large-margin online algorithm which has been employed successfully for a number of structured classification tasks in NLP, such as dependency parsing (McDonald et al., 2005). We employ the 1-best version of MIRA whose update rule is given by:

$$\begin{aligned} min \quad & ||w^{i+1} - w^i|| \\ s.t. \quad & s(x_t, y_t) - s(x_t, y') \geq L(y_t, y') \\ \forall y' \quad & \in best_1(x_t; w^{(i)}) \end{aligned}$$

On each update, MIRA attempts to keep the new weight vector as close as possible to the old weight vector, subject to *margin constraints* that keep the score of the correct output above the score of the guessed output by an amount given by the loss of the incorrect output. Similar to the perceptron, the averaged MIRA weight vector is used at test time.

4. Loss functions and Update strategies

Both the perceptron and MIRA algorithms require a feature vector representation of the correct output in order to update the weight vector. Ideally, we would like to be updating towards the feature vector representation of the gold standard output (surface form and alignment). However, the alignment information is hidden and only the gold standard surface form is visible. Even then, due to distortion limits and the way phrase-pairs are extracted, sometimes the reference surface form is just not reachable by the model. When the reference is reachable by the model, there might be multiple derivations to go from source to reference string. In this scenario in our current implementation, we pick the highest scoring derivation given the current model as the truth.
When the reference is not reachable, we investigate two alternate strategies for selecting the surrogate reference. Both strategies rely on selecting a reachable translation which is closest to the reference as measured by a *loss function*.

4.1. Loss function

As loss function, we have implemented sentence level smoothed BLEU (sBLEU) (Lin and Och, 2004) . Note that since we run our learning algorithm on the parallel training corpus, we only have one reference translation per source sentence. It is unclear how reliable a sentence level BLEU with respect to one reference is. Also while BLEU computes the brevity penalty by aggregating the length of the whole document, the sentence level BLEU computes the brevity penalty purely at the sentence level.
The MIRA update rule requires a loss function $\mathcal{L}(y_t, y')$ which indicates the penalty to be incurred when the guessed output $\mathbf{y'}$ is proposed instead of the correct output $\mathbf{y_t}$. This, in our work is given by:

$$L(y_t, y') = sBLEU(y_t) - sBLEU(y') \qquad (2)$$

4.2. Update strategy

The first update strategy we consider is one proposed in Tillmann and Zhang (2006). In this method which we call **max-BLEU update**, we modify our decoder to use sentence level smoothed BLEU as a scoring function. The decoder is run over the training data as a pre-processing step and the best scoring translation for each source is cached. Since we use approximate inference, search errors are possible. The cached translations are used as surrogate references in the online learning step. If at some point during learning the best guessed translation is found to have a higher BLEU than the cached surrogate, this guess becomes the surrogate from there on.

The second update strategy we look at is the **local update** proposed in Liang et al. (2006). At each decoding step, an n-best list is generated. The translation candidate in the n-best list with the lowest loss with respect to to the true reference is chosen as surrogate. While it is expected that the quality of translations in the n-best list improves over time as the weight vector estimate become better, this is not guaranteed to happen. Therefore, the surrogate from the previous iteration is merged with the current n-best list and the best translation from the expanded n-best list is chosen as new surrogate.

5. Features

Since one of the aims of this work is to show how much leverage we can get using purely discriminatively trained features, we eschew all probabilistic features such as language model, translation and reordering probabilities. Of course, one of the attractions of the discriminative framework is that we can throw in all kinds of features in the model, so in theory we could incorporate probabilistic features too. We leave this for future work.

We define five feature template classes :

- phrase pair
- source bigram
- target bigram
- orientation
- source distortion

Taking as example the derivation in Figure 1, an example of a source bigram feature would be:

$$f_{100}(b_i, b_j, o) = \begin{cases} 1 & \text{if } src(b_i) = \text{“morgen”} \\ & \wedge src(b_j) = \text{ich} \\ 0 & \text{otherwise} \end{cases}$$

An example of orientation feature is :

$$f_{1200}(b_i, b_j, o) = \begin{cases} 1 & \text{if } src(b_i) = \text{“morgen”} \\ & \wedge src(b_j) = \text{ich} \\ & \wedge tgt(b_i) = \text{“tomorrow”} \\ & \wedge tgt(b_j) = \text{i} \\ & \wedge o = DISCONTINUOUS \\ 0 & \text{otherwise} \end{cases}$$

The source bigram feature allows to capture source side reordering patterns. Similarly the target side bigram feature allows to capture fluency on the english side and is therefore a language model-like feature. The phrase-pair feature is a discriminative equivalent to the $p(e|f)$ generative feature.

To alleviate data sparseness for these lexical features, we use a rolling window of 3 words.

6. Experiments

We ran experiments on the Prague Czech-English Dependency Treebank (PCEDT) (Hajič, 1998) which consists of 21141 sentences from the Penn WSJ corpus translated into Czech and annotated with morphological information as well as dependency structure information. About 250 sentences each for development and test were translated once into Czech and then back into English by five diferent translators.

We decided to use this dataset for 2 main reasons : (a) it is a small corpus therefore allowing us to train several models in a short time and (b) it is richly annotated with linguistic infomation which we would like to incorporate as features of our model in future work. However, one of the major drawbacks of using the PCEDT is that since Czech is a morphologically very rich language and the training set is small, the corpus suffers from severe data sparseness issues (Goldwater and McClosky, 2005).

We obtained word alignments by using the GIZA++ toolkit (Och and Ney, 2003) on the training corpus in both translation directions. The two sets of alignments were then symmetrised using the **grow-diag-final** method previously described in (Koehn et al., 2005) and phrase-pairs consistent with the alignments were extracted. Note that both the perceptron and MIRA are **error-driven** algorithms in that parameter updates are performed only when the learner is unable to classify a training instance properly. Given that the same training data is being used for phrase extraction and for parameter estimation, overfitting is possible. We would instead like to make the training enviroment as hard as the testing one. Since Koehn et al. (2003) show that there is not much performance gain in using phrases of length longer than 3 words, we limit our phrase table to phrases upto 3 words long.

6.1. Comparing models

In a first set of experiments, we wanted to have an exact comparison between a purely discriminatve approach and a standard generative-hybrid model trained using the same amount of monolingual and parallel data.

Our hybrid phrase-based MT system uses the following feature functions:

- phrase translation probability (in both directions)
- lexical translation probability (in both directions)
- word penalty
- phrase penalty
- language model score
- linear reordering penalty
- lexicalised reordering model (Koehn et al., 2005) score

Table 1: BLEU score and length ratio on training for oracle (O) and guessed (G) translations for 1-best MIRA

Iteration	1	2	3
Local O	46.79 (0.95)	52.50 (0.95)	54.69 (0.95)
Local G	35.62 (0.95)	45.10 (0.95)	49.93 (0.95)
MaxBLEU O	60.31 (0.90)	60.43 (0.90)	60.51 (0.90)
MaxBLEU G	29.21 (0.91)	52.80 (0.91)	57.15 (0.90)

Training scheme	BLEU	Len ratio	#Feats
Hybrid-LM	**34.53**	0.978	14
Hybrid-NoLM	27.54	0.978	14
Perceptron -local	27.40	0.912	4.1M
MIRA - local	**27.47**	0.912	4M
Perceptron - MaxBLEU	25.21	0.881	5.7M
MIRA - MaxBLEU	24.63	0.889	6.4M

Table 2: Results on Czech-English

The extracted phrase pairs were assigned probabilities by unsmoothed relative frequency, and the translation probabilities were lexically weighted as in Koehn et al. (2003). The word and phrase penalties simply add a constant factor for each word or phrase generated, to bias the model towards longer or shorter output. The basic reordering model only considers the linear distance that a phrase needs to be moved in order to align with its translation. This movement distance is measured on the foreign side. The linear reordering penalty simply adds a cost factor, δ^n, for all movements over n words.

We used the SRI Language Modeling toolkit (Stolcke, 2002) to train a trigram language model on the English side of the corpus, and trained the weights used to scale the feature functions via MERT. We trained 2 different systems, one that uses a language model(Hybrid-LM) and one that does not (Hybrid-NoLM).

For the purely discriminative models, in addition to the discriminative features described in Section 5., we also incorporate a linear reordering penalty and a word penalty feature. Both models were run with a reordering limit of 6 and an n-best list of 1000.

The online learning algorithms were run over the training data for 10 iterations. Performace on the development set was evaluated at the end of each iteration, and the best performing model was used to decode the test set. We tried two update strategies - max-BLEU update vs local update.

We first present in Table 1 training results for two experiments - training the 1-best MIRA with local updates and training it using max BLEU updates.

The Max BLEU references are of better quality though the local update references quickly improve. Training performance rapidly increases across iterations, drastically in the case of max BLEU training - a case of possible overfitting. The length of the surrogate references is much shorter than the lengths of the true reference, much more so in the case of max BLEU, causing the model to learn to produce short translations. After only 3 iterations, the translations proposed by the model are already quite close to the references. Test results are presented in Table 2.

Performance using discriminative training are much worse than the hybrid model (Table 2). The local update strategy outperforms the max BLEU update significantly even though the quality of the max BLEU references during training is better (Table 1). This corroborates the findings in (Liang et al., 2006) that conservative updates towards a surrogate extracted from the n-best list are more effective than aggressively updating towards a high-scoring surrogate. There is not much difference between MIRA and perceptron though weighting the update by the quality of the surrogate seems to help. Both learning algorithms converge quickly (between 2 to 5 iterations).

One of the reasons for the poor performance is that our models are producing **systematically short outputs** (21.5 words/sentence vs 24.7 words/sentence for MERT) as indicated by the length ratio (output length/reference length) column in Table 2. This result is to be expected given the trend seen during training.

We believe that short surrogates are being selected because of the way we compute the sentence level BLEU. Even though we include a brevity penalty term in our loss function, the penalty is only computed at a sentence level. The sentence-level scores do not translate directly into the document-level score because in the latter case the brevity penalty is aggregated for the entire document set.

Another possible reason for poor performance is that out of the 262,188 phrase pairs in the phrase table, only around half are assigned non-zero weights during training. Recall that only phrase-pairs seen in the reference or the guessed solution get updated. Since we eschew probabilistic features in these experiments, there are no translation scores to back off to for almost half of the phrase table entries. One possible way to fix this would be to use word-based translation features similar in spirt to the lexical translation probabilities used in Pharaoh. Another would be to perform more updates per training instance. Since decoding is a costly operation, it would make sense to make as much use as possible of it. We could select **m** references instead of just **one** and update versus the **n** highest scoring solutions instead of the **1**-best solution. This could be incorporated in MIRA through the use of margin constraints.

6.2. Learning from noisy data

Both MIRA and the perceptron work by boosting features in the reference and penalizing features in the incorrect output. However, as a consequence of using surrogate references, it is often the case that the reference used is a noisy translation.

For example a surrogate reference might contain the target ngram *"the the"* which the learner would assume as the truth and thus promote. In the perceptron and 1-best MIRA algorithms, only features that are in the reference or in the guessed output get updated. If the noisy feature is relatively sparse and is not seen often in either the reference or the guessed output, its weight might remain high.

One solution to the problem would be to weight the parameter updates by the quality of the surrogate. If the surrogate is actually the true reference or something very close to it, the parameter updaes should be more confident than if the surrogate is of poor quality. Loss functions are the obvious

Training scheme	BLEU	Len ratio	#Feats
Hybrid-LM	**34.53**	0.978	14
Hybrid-NoLM	27.54	0.978	14
W Perceptron -local	**28.09**	0.906	4.0M
W MIRA - local	27.64	0.911	4.0M
W Perceptron - max BLEU	24.04	0.881	6.0M
W MIRA - max BLEU	24.04	0.881	6.0M

Table 3: Results on test data using the weighted learning algorithms.

way to implement this idea.
For MIRA, our modified loss function is :

$$L_{mod}(y_t, y') = (sBleu(y_t) - sBleu(y')) \cdot sBleu(y_t) \quad (3)$$

The perceptron update rule is similarly modified:

$$w^{i+1} = w^i + sBleu(y_t) \cdot (\Theta(x_i, y_t) - \Theta(x_i, y_g)) \quad (4)$$

Whenever the surrogate reference is equal to the reference translation, we are left with the standard MIRA and perceptron update rules.
Note that noisy references are not issue in MER training since the features used are (mostly) complex probabilistic models. On the other hand, the word-based model proposed by (Ittycheriah and Roukos, 2007) does use a fine-grained feature set but is able to to learn directly from the reference translation. We believe that this is one of the reasons behind its good perfomance.
We refer to these modified algorithms as weighted Perceptron and weighted MIRA respectively. Results are shown in Table 3.
The weighted learners improve performance in 3 out of the 4 learning setups suggesting that this is a promising avenue to explore.

7. Conclusions

In this paper, we investigated discriminative training of a phrase based SMT system using millions of features. We show that reasonable results can be obtained even without using any probabilistic features. Model parameters are learned through two different online learning algorithms, the perceptron and MIRA. Results show that there is not much to choose between the two; however, MIRA provides an elegant framework to incorporate loss functions and margin constraints. We also find that a conservative update strategy whereby a surrogate reference is picked from among an n-best list is better than updating aggressively towards the reference or a candidate close to the reference in cases where the reference is not reachable. We also presented variants to the perceptron and MIRA which help improve performance, by taking into account the 'goodness' of the surrogate reference in the update rule. This is a way to make the learning algorithms more robust to noisy references. Future work will look into improving the performance of the discriminative model by using more generalized feature sets such as POS tags and word classes as well as by incorporating probabilistic features such as a language model.

8. Acknowledgement

This work was supported in part under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022.

9. References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation. *Computational Linguistics*, 19(2):263–313.

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

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

Sharon Goldwater and David McClosky. 2005. Improving statistical mt through morphological analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 676–683, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Jan Hajič. 1998. Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In Eva Hajičová, editor, *Issues of Valency and Meaning. Studies in Honor of Jarmila Panevová*, pages 12–19. Prague Karolinum, Charles University Press.

Abraham Ittycheriah and Salim Roukos. 2007. Direct translation model 2. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 57–64, Rochester, New York, April. Association for Computational Linguistics.

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

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *International Workshop on Spoken Language Translation 2005*.

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 44th Annual Meeting of the Association for Computational Linguistics*, pages 761–768, Sydney, Australia, July. Association for Computational Linguistics.

Chin-Yew Lin and Franz Josef Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of Coling 2004*, pages 501–507, Geneva, Switzerland, Aug 23–Aug 27. COLING.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 91–98, Morristown, NJ, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2001. Discriminative training and maximum entropy models for statistical machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302, Morristown, NJ, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–52.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Morristown, NJ, USA. Association for Computational Linguistics.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*.

Christoph Tillmann and Tong Zhang. 2006. A discriminative global training algorithm for statistical mt. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 721–728, Sydney, Australia, July. Association for Computational Linguistics.