Discriminative Models and Training Methods For Statistical Machine Translation

Abhaya Agarwal

May 6, 2008

Abstract

Statistical Machine Translation (SMT) has been the dominant flavor of Machine Translation (MT) over the last decade. Traditional SMT systems have a pipeline structure in which different kinds of Machine Learning models are employed in different stages. For the translation modeling, most state of the art systems use hybrid models that combine a handful of generative models in a discriminative framework. The generative models are estimated over large amounts of parallel data and are used as features in a log-linear model which scales them to get good translation performance. Unfortunately, these models don't scale well with the number of features. As a result, the main advantage of discriminative models, the use of millions of arbitrary features capturing fine grained properties of data, is given up. One approach to solve this problem is n-best re-ranking on top of a base model, that has proved successful in Natural Language Parsing. Other option is to have purely discriminative translation models that can train directly on the parallel bilingual data and employ any arbitrary feature. This survey primarily focuses on the second approach but provides pointers to the recent work on both re-ranking and more traditional hybrid models. We cover the purely discriminative models described in literature and outline the major obstacles that must be overcome before these models can perform comparable and better than the current state of the art systems.

1 Introduction

SMT systems are the current state of the art in the area of Machine Translation. Starting with the IBM translation models 1-5 [Brown et al., 1990], the translation models in these systems have become increasingly powerful and well motivated. The parameters of these models are learned generatively, either with an iterative algorithm like EM or with simple surface heuristics. These models are then combined along with some other features in a log-linear framework which is a generalization of the Noisy Channel Model and allows for the inclusion of arbitrary features. The log-linear model scales these base models to get good translation performance on unseen data. The log-linear model is trained on a small amount of held-out data using Minimum Error Rate Training (MERT)[Och, 2003] which tries to directly maximize the evaluation metric of one's choice. The number of features in this setup is around 10-15 and it is difficult to scale up MERT to a large number of features.

On the other hand, discriminative models with big feature sets have become increasingly popular for NLP applications [Collins, 2000][McDonald et al., 2005]. These models allow the use of million of arbitrary, possibly overlapping, features and can be efficiently trained if the features are suitably localized. This provides a way to include any domain specific knowledge one might have without

worrying about the dependencies between the features. In addition, various kinds of regularization provide inherent feature selection mechanism that removes the need for large scale feature engineering and leads to sparser and manageable models.

As the MT systems get better, the errors committed by them become more subtle. So it is natural to hope that larger feature sets would provide the flexibility to model more fine grained aspect of translation process. There were earlier attempts in [Och et al., 2004] and [Shen et al., 2004] to do n-best re-ranking over a baseline systems output using a large feature set. More recently, work has been done on large scale discriminative training of translation models directly from bilingual parallel data (Liang et al. 2006, Watanabe et al. 2007, Ittycheriah and Roukos 2007, Arun and Koehn 2007). These systems have shown modest improvments over the more popular MERT trained hybrid models. Given the amount of data on which the modern MT systems are trained, it is a significant challenge to scale up the discriminative training but there are definite advantages in discriminative approach which would be good to have in SMT system building toolbox.

The rest of the report is organized as follows. In section 2 and 3, we briefly review some discriminative methods and translation models respectively. Section 4 presents the work done on the large scale discriminative training which is the main focus of this report. A quick overview and some pointers to the recent work on the currently used hybrid models and re-ranking approaches follows in sections 5 and 6. We end with a discussion of the main obstacles that are faced by the discriminative methods in MT and future directions in section 7.

2 Discriminative Models and Training Methods

The distinction between the generative and discriminative methods is not completely black and white, both in theory and practice and sometimes people may differ in what they call discriminative. Most of the times, the so called purely discriminative models will include features derived from simple generative models.

For the purpose of this report, we will classify models of the join probability of the input and output i.e. $p(y, x)$ and associated training methods as generative while all the models that treat input as given and hence do not model it, will be called discriminative. Note that we have said nothing about what the discriminative methods will model and how they will model it yet. The common theme binding all the discriminative methods is that they do not model the probability density of the input and treat it as given. So according to this definition, a HMM is a generative model since it models $p(y_1^n, x_1^m)$. On the other hand, a CRF would be a discriminative model since it models $p(y_1^n|x_1^m)$.

Discriminative methods primarily come in two flavors.

2.1 Likelihood Based

Likelihood based discriminative methods model the conditional density of the output given the input. Most popular models of this kind are conditional log linear models. Let x be the input and y be the output. if Φ is the feature vector and $\mathbf{w}$ is the corresponding weight vector, a conditional log linear model takes the following form:

$$p(y|x_i) = \frac{e^{\mathbf{w}\cdot\Phi(y,x_i)}}{\sum_{y_j \in G(x_i)} e^{\mathbf{w}\cdot\Phi(y_j,x_i)}}$$

The parameters of the log-linear model can be estimated using maximum likelihood estimation.

The likelihood function of log-linear models as defined above, is convex and so any gradient based method and be used to maximize the likelihood.

However, maximum likelihood is not the only method for estimating parameters. In machine translation, the final performance of the model is measured by some evaluation metric like BLEU[Papineni et al., 2001] and not the likelihood of the output. There is no reason to believe that the parameters tuned to maximize the likelihood of the data will also lead to optimal BLEU score. One option is to optimize parameters to maximize the BLEU score of the training data. However the error surface defined by BLEU will be non-convex in general and gradient based methods are not guaranteed to find the global optimal solution. Additionally, the sum in the denominator is over all the possible output sentences which is a exponentially large space. MERT [Och, 2003] is one way of optimizing the parameters so that the BLEU score of the training data is maximized. Section 5.1 on page 12 provides pointers to other loss functions and training methods useful for MT.

The sum in denominator is called partition function and ranges over all possible values of t. Computing the partition function is not necessary while decoding since it is constant for all ts but it is required at the time of training. The main challenge in the application of log-linear models is to be able to compute the partition function efficiently. The problem is worse in case of NLP where the sum is over structures. However, if the features are local, the sum can be obtained using well known dynamic programming techniques [1]. As a result, the features are often restricted to be local so that the partition function and the feature expectations can be efficiently computed [2].

2.2 Margin Based

Our primary aim in discriminative training is to find a model that performs well on the unseen data. One way of achieving is to let go the underlying density modeling in the likelihood based methods and directly look for a weight vector $\mathbf{w}$ such that for all the training examples

$$y_i = argmax_{y \in G(x)} \mathbf{w}.\Phi(x_i, y)$$

The basic idea is to find a hyperplane that separates the correct output from the others. SVM-struct, MIRA and Perceptron are some of the algorithms in this category. Perceptron and MIRA are examples of Online algorithms, a class of algorithms that look at one training sample at a time and update their weight vectors accordingly.

2.2.1 Perceptron

Perceptron is a simple and easy to train online learning algorithm. The Perceptron was extend for structured classification by [Collins, 2002] and has proved to be effective for a range of NLP tasks. In each iteration, it decodes the training examples one by one using the current weight vector and updates the weight vector with the difference in the feature vector of the current best and the gold standard. The final parameters are obtained by averaging the parameters across all iterations to avoid over-fitting [Collins, 2002] .

2.2.2 MIRA

MIRA [Crammer and Singer, 2001] uses the concept of a loss function that is used to scale the margin. A loss function tells us how much penalty we incur when we predict something wrong. A common

[1] Forward-Backward algorithm for sequences and Inside-Outside for tree structures

[2] Another option is to approximate the space of all the outputs with the n-best list

example is a zero one loss function where you score 1 if the output is correct and 0 otherwise.

The optimization problem solved by MIRA is the following. Let x_t be the t^{th} training example, w^i the weight vector in the i^{th} iteration, $\Phi(x_t, y)$ the feature vector representation of x_t, $G(x_t)$ the set of all the possible outputs corresponding to x_t and $L(y, y^{'})$ a given loss function, then

$$\begin{gathered} \min ||\mathbf{w}^{i+1} - \mathbf{w}^i|| \\ s.t. \quad \mathbf{w}^{i+1}.(\Phi(x_t, y) - \Phi(x_t, y^{'})) \geq L(y, y^{'}) \\ y' \in G(x_t) \end{gathered}$$

The idea here is to create a margin between the correct output and an incorrect output that is at least as big as the loss between them. When the number of possible output is huge, so is the number of constraints and solving this optimization problem would be hard. As an alternative, we can use the k-best version [McDonald et al., 2005] where the constraints are formed with only the k-best outputs. Intuitively the k-best outputs will be closer to the correct output and so satisfying the corresponding constraints would be more important. The k-best formulation would look like

$$\begin{gathered} \min ||\mathbf{w}^{i+1} - \mathbf{w}^i|| \\ s.t. \quad \mathbf{w}^{i+1}.(\Phi(x_t, y) - \Phi(x_t, y^{'})) \geq L(y, y^{'}) \\ y' \in best_k(x_t; w^i) \end{gathered}$$

Similar to Perceptron, the common practice is to use the average of the parameter vectors from all iterations to avoid over-fitting.

2.3 Other Algorithms

There are a lot of other discriminative learning algorithms and models available like SVM-struct and decisions trees. Not many of them have been applied to the problem of full scale machine translation yet. In this survey, [Wellington et al., 2006] use boosted decision trees on the MT sub-tasks of word and tree transduction (See section 4.2.2 on page 10).

3 Translation Models

In Statistical Machine Translation, given some amount of parallel text between source and target language, the aim is to learn a translation model that can translate unseen source language data into target language. For the purpose of this survey, we assume basic familiarity with the SMT methodology. [Lopez, 2007] provides a comprehensive survey of Statistical Machine Translation.

Although the translation models translate one sentence at a time, most of them record the bi-lingual translation equivalence in form of smaller pieces in order to get better generalization on unseen data. Phrase based models use phrases-pairs, IBM models use word-phrase pairs and Hiero style systems use phrase-pairs with variables. We call any such elementary piece a Basic Translation Unit (BTU).

Given a set of BTUs, a derivation under a translation model defines the steps involved in constructing the full sentence pair from individual BTUs. Under most translation models, there are more than one ways of constructing a given sentence-pair. This is called spurious ambiguity. If the underlying

model has a probabilistic interpretation, we can handle this ambiguity by summing over all the derivations. When the model doesn't have a probabilistic interpretation like in margin based discriminative models, it is a more tricky issue. We will return to this issue in section 4.1.4 on page 8.

Derivations are also important because they suggest a natural parametrization of translation models. For example, in a SCFG based translation model, one parameter per SCFG rule would be a natural parametrization of the model (It is not a requirement to parametrize the model along this structure but it would seem to be a natural choice).

We now briefly describe the phrase based and hierarchical phrase based translation models that are used in most of the work that we describe in section 4.

3.1 Phrase Based SMT (PBSMT)

Phrase based SMT models, consist of 3 steps. First the input sentence is segmented into phrases. A phrase in this model is any contiguous span of source sentence and has no linguistic meaning attached. Now each source phrase is translated into target language phrase and finally the target language phrases are reordered to generate the final output.

Bi-lingual phrase pairs are the basic unit of translation in this model. To extract them from the parallel data, one option is to use EM to directly estimate a phrasal alignment between two sentences. However, the total search space of phrasal alignments is huge. So as an alternative, phrasal alignments can be read off from the word alignments instead. Word alignments can be generated by any of the various word alignment methods available. Finally all the phrase pairs consistent with the word alignment are extracted. To keep the number of extracted pairs manageable, only phrases of up to a max-length are extracted. In practice, this heuristic method of phrase extraction has been shown to work better than the more principled EM based alignment and extraction.

In the current phrase based systems, every phrase pair is assigned 4 features, $p(s|t)$, $p(t|s)$, $p(s_{lex}|t_{lex})$, $p(t_{lex}|s_{lex})$ where $p(s|t)$ and $p(t|s)$ are relative frequency estimates of phrase-pair probability in two directions and $p(s_{lex}|t_{lex})$ and $p(t_{lex}|s_{lex})$ are lexically weighed translation probabilities of the phrase pair in two directions. A typical phrase based decoder uses these 4 features in combination with one or more language model scores, distortion penalty, phrase and word penalties in a log-linear combination to search for the best hypothesis. For more details, please refer [Koehn et al., 2003].

3.2 Hierarchical Phrase Based SMT

In vanilla PBSMT, the phrases can only have lexical items and must be contiguous. As a result, many kinds of language divergences seen in real translation data can not be modeled. Additionally, the reordering models used in these systems are distance based and not very strong. Hierarchical phrase based model proposed by [Chiang, 2005] extends the phrase based models by allowing phrases to contain variables that can be replaced with other phrases. This can be alternatively seen as a single variable SCFG induced from translation data. Features used in this model are similar to the vanilla phrase based model. Decoding in this model is equivelent to parsing the source side with the induced grammar while genarating the target sentence alongside. For reasons of efficiency, phrases can only have upto 2 variables in them which should not be contiguos. With this restriction, the induced grammar is in Chomsky Normal Form and can be efficiently parsed using an extension of CKY algorithm.

For the exact method of extraction and details about decoding, please refer [Chiang, 2007].

4 Large Scale Discriminative Models for MT

In this section, we first describe the issues that a discriminative SMT system must address. We then present some specific examples. Performance of all the systems discussed in this sections has been summarized in the Table 6 on page 18.

4.1 Issues in a large scale discriminative SMT system

In order to use discriminative techniques in a machine translation system, several issues need to be addressed. The first question to ask would be if all kinds of translation models can benefit equally from discriminative techniques. No such comparative results are available in the literature as of now. Most of the work in the area has used one of the two models described in section 3 on page 4, with some interesting variations [Ittycheriah and Roukos, 2007] [Wellington et al., 2006].

For any translation model, the process of obtaining the repository of BTUs is the same as hybrid SMT systems. There has not been an attempt to engineer a single, purely discriminative system from the parallel data itself. Both large scale discriminative and hybrid systems follow a two layer approach where first a word alignment model is used to align the data and then basic translation units are extracted from it[3].

4.1.1 Features

The main attraction of using a discriminative approach is the large number of arbitrary features that can be used in the model. Features can be overlapping and can be either binary or real valued. They can encode any information present in the full input and output sequences and also the associated derivations. However the training and decoding become increasingly expensive as the features become non-local[4]. The primary criteria for choosing features is the balance between usefulness of features and efficiency of training/decoding.

The only large scale experimentation on a variety of features for machine translation reported in the literature is [Och et al., 2004] which was done in a re-ranking setting. The features tried varied from lexical features to various syntax based translation models proposed in the literature till that time. The results were pretty disappointing with no feature other than IBM model 1 scores showing significant improvement over baseline. The authors list many possible reasons for this result including the problem in getting reliable annotations like POS tags, parse trees on the MT output which is not well formed.

Later researchers have tried a more vanilla set of features but with slightly more success. A natural feature to include in these systems is the one marking the presence of a particular BTU in building the source-target pair [Blunsom et al., 2008] [Arun and Koehn, 2007]. One such feature fires corresponding to each of the millions of BTUs available. We can also look at the target language bi-grams that capture the fluency of the output. Source side bi-grams can help in encoding re-ordering patterns.

If one preserves the translation correspondences inside a BTU, useful features that tie across the BTUs can be used. An example would be word pair features similar to phrase pair features. [Liang et al., 2006] used a alignment constellation feature that marks the presence of particular alignment pattern within a phrase pair. The interesting finding there was that the long monotonic patterns received the lowest weights while word inversion patterns were at the top. This suggests that the in case of monotonic patterns, many smaller phrase pairs are preferred over one long one.

[3] [May and Knight, 2007], also see section 7.1.2 on page 13

[4] Since the input is treated as given, features can draw on the full input sequence without being non-local

Another source of features for discriminative models are the base generative models. Experience in the word alignment task has been that state of the art results are achieved when using the generative model predictions as features [Taskar et al., 2005]. Experiments in [Arun and Koehn, 2007] and [Blunsom et al., 2008] suggest a similar case for SMT also. With only the discriminative features, they fail to beat the hybrid MERT baseline. [Liang et al., 2006] and [Watanabe et al., 2007] on the other hand, use the translation model scores and language model scores from generative models as features and report improvements over the MERT baseline.

4.1.2 Training methods

There are a host of discriminative training methods available, some of them described in section 2. There is no conclusive evidence about which method works best for MT. The choice of the training method is primarily driven by the kind of features in use and the efficiency of the whole process since discriminative training is computationally expensive.

Among the margin based methods, [Liang et al., 2006] used the basic Perceptron algorithm along with a local update strategy while [Watanabe et al., 2007] employed MIRA along with a improved version of local update strategy, described in section 4.1.4 on the following page. [Arun and Koehn, 2007] present a comparison of Perceptron and MIRA and find no significant differences in performance.

The learning algorithm of [Tillmann and Zhang, 2006] is also inspired by margin based methods. It tries to maximize a cost sensitive margin between a set of good translations and other alternatives. The set of alternatives in build up by including the 1-best output of the decoder in each iteration.

[Blunsom et al., 2008] model $p(t|s)$ directly using a global conditional log linear model in a Hiero Style system. There is one feature per rule in the model. They estimate the model using MAP estimation which maximizes the likelihood of the training data penalized with a prior. The prior is a zero mean Gaussian. They use a packed forest representation of all the derivations produced by the model and compute the partition function and feature expectations using inside-outside algorithm.

4.1.3 Loss functions

Some discriminative learning algorithms e.g. MIRA require a loss function as part of their update rule (Section 2.2.2 on page 3). Since the final translation performance is often measured using BLEU metric, most of the work on discriminative training of MT models uses a loss function based on it. Originally BLEU is computed by aggregating the statistics over the whole corpus but loss function needs to be computed at the sentence level. An alternative is the smoothed BLEU or sBLEU, a segment level modification of the original metric.

[Arun and Koehn, 2007] use the difference in sBLEU scores of the reference and the hypothesis. They also experimented with weighing the difference with the absolute sBLUE score of the reference. This is prompted by the fact that some of the reference translations are not gold standard translations but surrogates obtained from a n-best list[5]. As expected , the weighted loss function performs better.

[Tillmann and Zhang, 2006] use the following function of sBLEU scores and the translation model scores where b and s are the BLEU score and translation model scores of the reference and b' and s' of the hypothesis.

$$\phi(s, b; s^{'}, b^{'}) = (b - b^{'})(1 - (s - s^{'}))^2$$

This is a convex function which can be optimized using standard gradient based techniques.

[5] See section 4.1.4 on the following page

The main problem in using sBLEU as a loss function is that the brevity penalty of the original metrics is computed at a sentence level. This can prove to be too strict as evidenced by [Arun and Koehn, 2007] who noted that the discriminatively trained model was producing consistently smaller hypothesis. In [Watanabe et al., 2007], authors use a approximate BLEU formulation which avoids this problem by still computing BLEU at the sentence set level. The basic idea is to compute the BLEU score at the corpus level once with all the original references and once with one of the references replaced with the corresponding hypothesis. The difference in the scores is the loss accrued by the hypothesis at the corpus level.

4.1.4 Update strategies

Discriminative learning methods learn the model by updating towards a gold standard. In machine translation, the target side of parallel corpus is the gold standard. However in the real world MT systems, there are two immediate issues.

- The coverage of even large scale MT systems is not 100% on the training data because of the restrictions imposed on the underlying extraction processes. For many source sentences, the specified gold standard will not be present in the output space of the model. Updating towards a unreachable output is not desirable.

If the amount of unreachable training data is small, we can simply discard it but experience shows that this percentage can be very high [6]. The other option is to compute a surrogate reference that is reachable by the model and can be used in place of gold standard. [Tillmann and Zhang, 2006] use a modified decoder to generate the highest BLEU scoring hypothesis reachable by the model for every source sentence prior to training and use it as gold standard. These are called the MAX-BLEU references. On the other hand, [Liang et al., 2006] choose the surrogate reference as the highest BLEU scoring hypothesis from the n-best list generated by the decoder at each iteration of training. This is called a local update strategy. Improving on it slightly, [Watanabe et al., 2007] and [Arun and Koehn, 2007] keep around the surrogate references used in previous iterations and merge them with the current n-best list before choosing the new surrogate reference. [Arun and Koehn, 2007] present a comparison of two strategies.

- When the reference translation is reachable by the model, there are often more than one derivations. Some of these derivations are good and we want to reward them while others might be bad and we want to penalize them. However since the training data is not annotated with the gold standard derivations, we need some strategy to handle the ambiguity.

A compromise solution is to choose a derivation that is best according to some metric. [Arun and Koehn, 2007] use the best scoring derivation under the current model while [Liang et al., 2006][Watanabe et al., 2007] use the structure that comes with the highest BLEU scoring hypothesis from the current n-best list. The principled way of handling this problem would be to marginalize out the derivation by summing over all the possible derivations. However that might be too expensive in some models.

In [Blunsom et al., 2008], authors use a approximate beam search and sum over all the derivations that end up in the beam. They compare it against the compromise solution of choosing the derivation with maximum number of rules (based on the intuition that the smaller rules should be preferred in the system) and show significant improvements (Table 1) .

[6] 66% in [Liang et al., 2006] and 24% in [Blunsom et al., 2008]

System	BLEU Score
Discriminative max derivation	25.78
Hiero with reduced features	26.48
Discriminative sum derivations	27.72

Table 1: Effect of summing over all the derivations vs choosing one [Blunsom et al., 2008]

[Ittycheriah and Roukos, 2007] overcome this problem by first restricting the space of allowed basic translation units to 1-n phrases and then using a word alignment model to get the gold standard alignments between the source and the target.

- Every source sentence can have multiple correct translations. So even if the output of the system might differ from the gold standard, it may not be a bad translation. We would like to avoid penalizing these sentences.

Unfortunately, there is no straight forward way of achieving this. A partial mitigation can be achieved by using local update strategy even in those cases when the gold standard is reachable. [Liang et al., 2006] compare the aggressive, local and hybrid updating strategies and show that the local strategy works best.

4.2 Global vs Local Models

All the models that we have presented till now, tackle the translation problem globally i.e. all the decisions in the translation of one input sentence are taken jointly. A different approach involves breaking down the global decision process into a series of local decisions. This involves making independence assumptions and may not lead to globally optimal solutions. On the other hand, training and decoding in these models are much simplified. [Ittycheriah and Roukos, 2007] and [Wellington et al., 2006] are two examples of this approach.

4.2.1 Local conditional log-linear models

In [Ittycheriah and Roukos, 2007], the process of phrase based translation is decomposed into following steps:

- Begin at the left edge of source and consider a window of predefined size.
- Choose a jump size j and jump that many places in the source sentence (-5 $<$ j $<$ 5)
- Produce the target side corresponding to this source word and mark it as covered.
- Iterate till all source words are covered.

So the global models factors down as following:

$$p(T, j|S) = \prod_i p(t_i, j|s_i)$$

Each individual p(ti,j|si) is modeled as a mixture of language model score and a translation model score where the translation model is a conditional log linear models of the form:

Feature Name	Feature Variables
SRC_LEFT	source left, source word, target word
SRC_RIGHT	source right, source word, target word
SRC_TGT_LEFT	source left, target left, source word, target word
SRC_TGT_LEFT_2	source left, target left, target left 2, source word, target word

Table 2: Lexical Context Features Used in [Ittycheriah and Roukos, 2007]

$$p(t_i, j|s_i) = \frac{p_0(t, j|s)}{Z} \exp \sum_i \lambda_i \phi_i(t, j, s)$$

$p_0(t, j|s)$ is a prior distribution set to the normalized phrase count in this case. The parameters are estimated using Improved Iterative Scaling (IIS) [Della Pietra et al., 1997]. The features in the model include word pair features, lexical context features (Table 2), Arabic segmentation features formed by grouping every morpheme of the source word with the target side, POS features and coverage features that check if the surrounding source words have been already translated.

Compared to a phrase based baseline, the system performance was comparable or better depending on the test set. This is despite the fact that the discriminative system had a much more restricted set of BTU available as compared to phrase based system. Most of the gains come from the lexical coverage features while segmentation features also provided a little boost. Gains achieved from POS features and coverage features were not significant.

4.2.2 Word and tree transduction using multi-class classifiers

[Wellington et al., 2006] present another model along the similar lines. They do not address the end to end translation problem but experiment with the sub-tasks of word transduction and tree transduction. Tree transduction is the problem of predicting target tree given the source tree. They decompose this problem into a set of multiclass classification problems. One type of problems predict the word level translations of source words while others predict the internal nodes of the target tree given the source tree. A $l1$ regularized boosted decision tree is trained for each of these problems.

They present an interesting point about $l1$ vs $l2$ regularization. On the word transduction task, the model trained with $l2$ regularization was 2 orders of magnitude bigger than the model trained with $l1$ regularization. The difference in performance was small. However due to the large size, it was not possible to train the model with $l2$ regularization on the large amount of training data and a $l1$ regularized model trained on more data significantly out-performes the original $l2$ regularized model (Table 3). They conclude that for the problems of the scale of MT, we need regularization schemes that lead to sparse solutions and hence $l1$ regularization should be preferred.

Regularization	Performance (% accuracy)	Size (# of non-zero features)
$l1$(10k)	54.13	41.7k
$l2$(10k)	54.53	2.51M
$l1$(100k)	62.42	703k

Table 3: Effect of different regularizations on the task of word transduction [Wellington et al., 2006]

System	Filtering Technique	Number of Sentences in Training Data
Arun and Koehn [2007]	-	21k
Liang et al. [2006]	Length 5-15	67k
Ittycheriah and Roukos [2007]	For every n-gram that occurs in test set, select the first 20 sentences containing it	197k/267k/279k
Tillmann and Zhang [2006]	Keep sentences from the training data that have at-least one n-gram from test data	230k
Watanabe et al. [2007]	-	1k
Blunsom et al. [2008]	Length 5-15	130k

Table 4: Data Filtering applied to training data

4.3 Scaling up the discriminative training

The discriminative training is computationally expensive. Hence scaling it up to large amounts of data remains a big challenge. An average large scale SMT system is trained on more than 1 million sentence pairs [7]. On the other hand the systems described above were trained on far less data (67k [Liang et al., 2006], 21k [Arun and Koehn, 2007], 280k [Ittycheriah and Roukos, 2007], 130k [Blunsom et al., 2008]). Even for training on these small amounts of data, these systems need to use various heuristics. Table 4 shows a quick summary of the data filtering employed in different systems.

In the word alignment, the experience has been that as the amount of training data increases, the gains shown by more sophisticated models over the simpler models tend to disappear. Since the discriminative models still have to show significant improvement on state of the art systems, it is hard to predict what the results will be with large amounts of data.

5 Hybrid Models

Although showing promising trends, none of the purely discriminative methods have been scaled up for use in a large scale MT systems yet. So while these methods catch up, it is still important to see

[7] In WMT2008 translation shared task, the training data was about 1.2 million sentences

how the hybrid methods currently in use can be improved. There are two primary directions for such investigations, a better estimation technique and a more effective set of features. [Och et al., 2004] explored a wide variety of features most of which failed to give any significant improvement. [Nguyen et al., 2007] discuss the use of non-parametric features in SMT.

5.1 Improving the estimation

Currently, MERT is the standard procedure for optimizing the parameters of the log-linear combination of generative models while BLEU is the most frequently used criteria. In [Zens et al., 2007], authors evaluate many different training criteria including maximum likelihood at sentence and n-gram level and BLEU computed at sentence and n-gram level. They optimized the parameters using the Downhill Simplex algorithm. The parameters tuned to optimize n-gram level BLEU gave the best results on the two unseen test sets.

[Smith and Eisner, 2006] present Minimum Risk Annealing to estimate the parameters of a log-linear model against a arbitrary loss function. The idea is to minimize the expected loss instead of the 1-best loss. The expected loss is computed under a probability distribution which starts out flat over the hypothesis space but is concentrated more and more on the 1-best hypothesis as the training progresses. They compare their method against MERT and show significant and consistent improvements in BLEU score.

6 Discriminative Re-ranking

Another popular way of taking advantage of discriminative methods is to use them in a re-ranking setting. First of all, a n-best list is generated by a baseline system which uses a limited set of features that allow for efficient training and decoding. Then a discriminative re-ranker is used to re-rank the n-best list by making use of a large number of global features that capture long distance dependencies and other non-local features. Discriminative re-ranking has been quite successful in Natural Language Parsing.

[Shen et al., 2004] present 2 algorithms for MT re-ranking. The first is a splitting algorithm that tries to find a hyperplane separating the top r translations from the bottom k translations. The second algorithm is ordinal regression with uneven margins, proposed in [Shen and Joshi, 2004]. The feature used were derived from the set used in [Och et al., 2004]. The results showed no significant improvements over the MERT baseline.

7 Future Directions

The use of large scale discriminative models in MT has started only recently. Their performance levels are still behind hybrid systems[8] and only slightly better when using generative models as features. There are a number of research issues that need satisfactory answers before these models can become a serious challenge to current models.

Discriminative models have been used in various NLP tasks with reasonable success in the past. However, MT presents some unique challenges which broadly fall into two categories.

[8] Even when trained on same amount of data

7.1 Efficiency and scaling issues

It is well known that discriminative training methods are computationally intensive. This is because during training, they need to decode the entire training set once per training iteration. One of the reasons for the popularity of the Perceptron like algorithms is that they only require computation of 1-best hypothesis in each iteration. However, it is possible to generate k-best lists with very small cost on top of 1-best [Huang and Chiang, 2005]. This means that MIRA style algorithms that can exploit the top k outputs and make more informed decisions can be used at almost the same cost as Perceptron style algorithms[9]. Going a step further, one of the main reasons of the failure of n-best re-ranking to give significant gains is because the n-best lists often do not have enough variety among the candidates. Recently, state of the art results were obtained for parsing by applying re-ranking directly to a packed forest representation [Huang, 2008]. The same approach may give improvements for MT also.

7.1.1 Syntax vs Phrase based models

Another possible direction of investigation would be to see if some translation models are more suited for discriminative approaches than others. It is known that for SCFG and other syntax based translation models, decoding is polynomial in sentence length while for phrase based models, it is NP-Hard [Lopez, 2007]. In practice, phrase based models are usually faster because of the arbitrary limits imposed on the phrase sizes and reordering windows. It would be interesting to see if this polynomial time guarantee can be exploited to make discriminative training efficient and thus scale up to larger corpora.

7.1.2 Discriminative BTU extraction

A similar direction is to explore more restricted versions of the current translation models. [Ittycheriah and Roukos, 2007] restricted the underlying phrase based model to only use 1-n phrases but they did not directly tie up the decoding process with the phrase extraction. On the other hand, [Liang et al., 2006] explored alignment contellation features that give you some information about which alignment patterns are useful. A natural next step would be to tie the phrase extraction and translation model training more tightly together[10]. The feature selection that comes for free with some of the discriminative techniques can be useful here.

7.2 Problems with gold standard data

This is the issue we touched upon in section 4.1.4 on page 8. In many NLP tasks, there is a fixed gold standard known apriori and it is easy to build models such that the gold standard is always guaranteed to be reachable. Both these luxuries go away in MT and as we saw, current systems have explored some ways of handling these issues.

One interesting direction for exploration would be, given a set of gold standard sentences, how can we create a enlarged space of acceptable translations and how can we efficiently compute the similarity between this set and our candidate translation. In some sense, BLEU does the same thing by projecting all the reference sentences into a common space of n-grams against which the candidates are evaluated.

[9] We still need to solve the optimization problem at each step but the training times are currently dominated by the time spent in decoding.

[10] Similar experiments in generative models do not give better performance [DeNero et al., 2006][May and Knight, 2007]

However this is a low precision and low recall space (in terms of the acceptable translations of the source sentence) . [Pang et al., 2003] provide another example in the context of paraphrase generation but it is not known how efficient it would be to evaluate candidate hypothesis against such a space.

8 Conclusion

In this survey, we covered some recent work in employing discriminative modeling and training algorithms to SMT with large number of features. The experience from other tasks in NLP and the initial results from the limited data scenario look promising. However significant issues regarding scaling up of the techniques to large amounts of training data remain which must be addressed before that promise can be realized. The increasing popularity of syntax based models in which decoding in more manageable, can help with this problem.

Application of these techniques to SMT also presents some unique challenges such as the lack of one fixed gold standard. Future work can lead to development of newer discriminative techniques suitable for these tasks which will be useful for other fields as well.

9 Acknowledgments

The author would like to thank Shay Cohen for clarifying some points about the discriminative techniques.

References

Abhishek Arun and Philipp Koehn. Online learning methods for discriminative training of phrase based statistical machine translation. In *MT Summit XI*, 2007.

Phil Blunsom, Trevor Cohn, and Miles . Osborne. Discriminative synchronous transduction for statistical machine translation. In *To Appear in ACL 2008*, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990. URL `http://citeseer.ist.psu.edu/brown90statistical.html`.

David Chiang. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228, 2007. ISSN 0891-2017. doi: http://dx.doi.org/10.1162/coli.2007.33.2.201.

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P05/P05-1033`.

Michael Collins. 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, 2002. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1118693.1118694.

Michael Collins. Discriminative reranking for natural language parsing. In *Proc. 17th International Conf. on Machine Learning*, pages 175–182. Morgan Kaufmann, San Francisco, CA, 2000.

Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. In *14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001, Amsterdam, The Netherlands, July 2001, Proceedings*, volume 2111, pages 99–115. Springer, Berlin, 2001.

Stephen Della Pietra, Vincent J. Della Pietra, and John D. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

John DeNero, Dan Gillick, James Zhang, and Dan Klein. Why generative phrase models underperform surface heuristics. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 31–38, New York City, June 2006. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W06/W06-3105`.

Liang Huang. Forest reranking: Discriminative parsing with non-local features. In *To Appear in ACL 2008*, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Liang Huang and David Chiang. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 53–64, Vancouver, British Columbia, October 2005. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W05/W05-1506`.

Abraham Ittycheriah and Salim Roukos. 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 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N/N07/N07-1008`.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 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, 2003. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1073445.1073462.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 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 2006. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P06/P06-1096`.

Adam Lopez. A survey of statistical machine translation. Technical report, 2007.

Jonathan May and Kevin Knight. Syntactic re-alignment models for machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 360–368, 2007. URL `http://www.aclweb.org/anthology/D/D07/D07-1038`.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 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, 2005. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1219840.1219852.

Patrick Nguyen, Milind Mahajan, and Xiaodong He. Training non-parametric features for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 72–79, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W07/W07-0210`.

Franz Josef Och. 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, 2003. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1075096.1075117.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. A smorgasbord of features for statistical machine translation. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 161–168, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

Bo Pang, Kevin Knight, and Daniel Marcu. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *HLT-NAACL*, 2003.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA, 2001. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1073083.1073135.

Libin Shen and Aravind K. Joshi. Flexible margin selection for reranking with full pairwise samples. In *IJCNLP*, pages 446–455, 2004.

Libin Shen, Anoop Sarkar, and Franz J. Och. Discriminative reranking for machine translation, 2004.

David A. Smith and Jason Eisner. Minimum-risk annealing for training log-linear models. In *Proceedings of the International Conference on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL), Companion Volume*, pages 787–794, Sydney, July 2006.

Ben Taskar, Simon Lacoste-Julien, and Dan Klein. A discriminative matching approach to word alignment. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 73–80, Morristown, NJ, USA, 2005. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1220575.1220585.

Christoph Tillmann and Tong Zhang. A discriminative global training algorithm for statistical mt. In *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 721–728, Morristown, NJ, USA, 2006. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1220175.1220266.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 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, 2007. URL `http://www.aclweb.org/anthology/D/D07/D07-1080`.

Benjamin Wellington, Joseph Turian, Chris Pike, and I. Dan Melamed. Scalable purely-discriminative training for word and tree transducers. In *Proceedings of the 7th Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

Richard Zens, Sasa Hasan, and Hermann Ney. A systematic comparison of training criteria 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 524–532, 2007. URL `http://www.aclweb.org/anthology/D/D07/D07-1055`.

System	lang pair	Translation Model	Discriminative Technique Used	Baseline	Performance	Remarks
Liang et al. [2006]	Fr-En	PBSMT	Perceptron	28.8	29.6	
Watanabe et al. [2007]	Ar-En	Hiero Style	MIRA	49.33 47.03	49.81 48.41	
Tillmann and Zhang [2006]	Ar-En	PBSMT, block sequence model	CostMargin	-	35.9	On the same set, their earlier paper reported 37.8
Ittycheriah and Roukos [2007]	Ar-En	PBSMT	Local conditional log-linear model	51.20 49.06 36.92	51.19 50.00 38.61	1-n blocks only, translation process divided into local decisions
Arun and Koehn [2007]	Cz-En	PBSMT	MIRA/Perceptron	-LM : 27.54 +LM : 34.53	28.09	
Blunsom et al. [2008]	Fr-En	Hiero Style	Global conditional log-linear model	-LM : 28.14 +LM : 32.0	27.72	purely discriminative, one feature per rule used, sum over all the derivations that outperforms choosing one derivation

Table 6: Performace of various systems covered in this survey