経験的リスク最小化に基づいた統計的機械翻訳システムの最適化
Minimum empirical risk Training in Statistical Machine Trainslation Systems

林 克彦 † 渡辺 太郎 †† 塚田 元 †† 磯崎 秀樹 ††
Hayashi Katsuhiko† Watanabe Taro†† Tsukada Hajime†† Isozaki Hideki††
† 同志社大学 ††NTT コミュニケーション科学基礎研究所
†Doshisha University ††NTT Communication Science Laboratories

1 はじめに

統計的機械翻訳 (SMT: Statistical Machine Translation) では異言語間の翻訳作業を確率モデルで表現し，そのモデル尤度が最大となる解を探索することで翻訳を行う．従来，SMT では Noisy Channel Model において事後確率をベイズの定理から条件付確率と事前確率とに分割し，それらの積から尤度推定が行われてきた(1)．しかし，近年ではベイズの定理による枠組みを一般化した Log-linear モデルが主流となっている(2)．Log-linear モデルは機械翻訳を特徴付ける複数の素性とその重みを線形結合した形で表現される．Log-Linear モデルにおいて精度の高い翻訳を行うためには素性に係る重みを適切に決めることが重要となる．

重み最適化のアプローチとしてはじめに，最尤推定による手法が提案されたが(2)，未知の入力に対してより実質的な学習基準を用いた識別学習によるアプローチも提案されている (MERT: Minimum Error Rate Training)(3)．MERT ではシステム評価尺度である BLEU(4)や Word Error Rate(WER) などを学習基準として用いることができるため，最尤推定による手法よりも実質的な精度の面においてより有効な手法であることが示されている(3)．

一方，機械学習分野で開発された多くのアルゴリズムは経験的リスク最小化 (Empirical Risk Minimization) に基づいて定式化される．経験的リスク最小化とはある事象に対する有限の訓練データの確率分布から生じる誤識別の確率を最小化する学習基準である．MERT による学習もこの学習基準に基づいているが，目的関数として BLEU の値のみを考慮していることから学習法としてはやや特殊な定式化となっている．本稿では経験的リスク最小化に基づく学習法の 1 つである Support Vector Machine(SVM)(5) の定式化を取り入れ，最大マージン原理に基づいた SMT のための学習手法を提案する．提案法は経験的リスク最小化の標準的な定式化にならっていることから，MerT(Minimum empirical risk Training) と命名することにする．提案法の目的関数は BLEU 値だけでなく正則化項も考慮しているため，過学習の緩和が期待できる．また，提案法のパラメータ推定は，MERT の最適化法がそのまま利用できるため，非常に効率的である．

以下，第 2 節では SMT における Log-linear モデル，第 3 節では MERT による学習の枠組みについて説明を行う．第 4 節では本稿で提案する SVM をベースとした学習手法 (MerT) を定式化し，第 5 節において MERT と提案手法による実験結果を示す．第 6 節では実験結果を踏まえた上でまとめを行う．

2 統計的機械翻訳における Log-linear モデル

統計的機械翻訳では入力文 $\mathbf{f}$ を出力文 $\mathbf{e}$ に翻訳する作業を

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \Pr(\mathbf{e}|\mathbf{f})$$

とし，条件付確率が最大となる解を探索する問題として定式化される．Log-linear モデルはこの条件付確率を

$$\begin{aligned}\Pr(\mathbf{e}|\mathbf{f}) &= p_{\mathbf{w}}(\mathbf{e}|\mathbf{f}) \\ &= \frac{\exp[\sum_{m=1}^{M} w_m h_m(\mathbf{e},\mathbf{f})]}{\sum_{e'} \exp[\sum_{m=1}^{M} w_m h_m(\mathbf{e}',\mathbf{f})]}\end{aligned}$$

としてモデル化する．これは M 種類の素性関数 h を線形結合重み w で結合し，全ての解候補 $\mathbf{e}'$ によって正規化された形となっている．デコーダの過程において正規化項は必要ないため，次式で表す定式化となる．

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \langle \mathbf{w}, \mathbf{h}(\mathbf{e},\mathbf{f}) \rangle$$

3 Minimum Error Rate Training(MERT)

MERT による結合重みの最適化では出力文と参照訳から計算される損失関数を最小化するように学習を行う．MERT では損失関数以外の学習基準として BLEU のような評価尺度を用いることもできる．一般に SMT では BLEU を基準として学習を行うため，以下では BLEU を学習基準として定式化を行う．

通常，BLEU は翻訳結果と参照訳間における N-gram の重なりを corpus 単位でカウントして幾何平均する尺度である．入力文 $\mathbf{f}$ と参照訳 $\mathbf{r}$ から成る訓練データを $\mathbf{T} = \{(\mathbf{f}_1,\mathbf{r}_1),\ldots,(\mathbf{f}_S,\mathbf{r}_S)\}$ とし，訓練データ $\mathbf{f}_s$ に対する K-best 出力の集合を $\mathbf{C}_s = \{\hat{\mathbf{e}}_{s,1},\ldots,\hat{\mathbf{e}}_{s,K}\}$ とすると，MERT の学習基準は

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}}\, \mathbf{BLEU}\Big(\Big\{\mathbf{r}_s, \underset{\hat{\mathbf{e}}_s \in \mathbf{C}_s}{\operatorname{argmax}} \langle \mathbf{w}, \mathbf{h}(\hat{\mathbf{e}}_s,\mathbf{f}_s)\rangle\Big\}_1^S\Big) \qquad (1)$$

として表すことができる．

最適化法では Och によって提案された line-search(3)を用いる．この手法はパラメータのある 1 次元以外を全て固定し，固定していない 1 次元の最適化を行う手法である．まず，M 次元ベクトル $\mathbf{d}$ を定義する．ただし，ベクトル $\mathbf{d}$ は最適化を行う次元のみが 1 で他の次元は全て 0 をとる．これを用いて最適化を行う次元の変化分である α を最適化する問題に置き換える．

$$\begin{aligned}\hat{\mathbf{e}}_{s,best} &= \underset{\hat{\mathbf{e}}_s \in \mathbf{C}_s}{\operatorname{argmax}} \langle \mathbf{w} + \alpha\mathbf{d}, \mathbf{h}(\hat{\mathbf{e}}_s,\mathbf{f}_s)\rangle \\ &= \underset{\hat{\mathbf{e}}_s \in \mathbf{C}_s}{\operatorname{argmax}} \Big\{ \underbrace{\langle \mathbf{w}, \mathbf{h}(\hat{\mathbf{e}}_s,\mathbf{f}_s)\rangle}_{intercept} + \alpha \underbrace{\langle \mathbf{d}, \mathbf{h}(\hat{\mathbf{e}}_s,\mathbf{f}_s)\rangle}_{slope} \Big\}\end{aligned} \qquad (2)$$

これより各 K-best の解は横軸が α，縦軸が Log-linear モデルによる尤度 ($Score$) の直線として定義することができる．これを図 1 に示す

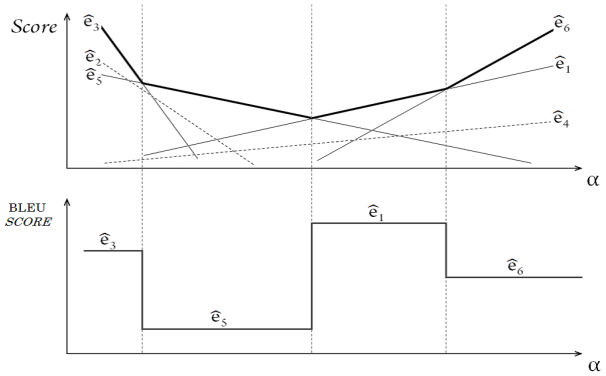

図 1　6-best による BLEU 値の局面

図 1 は 6-best 出力の集合で BLEU 値の局面を描いたものであり，直線が各解を表している．α の値が変わることによって $Score$ が最も高い解 (1-best) も変化し，この 1-best から BLEU の値を計算することで BLEU を最大とする α が求まる．図 1 の上図は式 (2) の argmax，下図が式 (1) の max の作業に相当する．

この手法では最適化を行っている次元に対してはグローバルな最適解が保証される．しかし，最適化の際に他の次元を考慮していないため，パラメータ全体においてはグローバルな最適解が保証されない．そこで最適化を行うパラメータの初期点を乱数によって複数点置くことで探索を行い，パラメータ全体が局所解に陥ることを避ける[7]．詳細は以下の疑似コードに示す．

Algorithm 1. Minimum Error Rate Training(MERT)

input initial parameter $\mathbf{w}_0$
input S-size training data $\mathbf{Tr} = \{\mathbf{C}, \mathbf{T}\}$
generate X random parameters $\mathbf{w}_1, \ldots, \mathbf{w}_X$
for $x = 0, \ldots, X$ **do**
  $\mathbf{w}^* = \mathbf{w}_x$
  $b\_scr^* = \mathbf{BLEU}(\{\mathbf{r}_s, \mathrm{argmax}_{\hat{\mathbf{e}}_s \in \mathbf{C}_s} \langle \mathbf{w}^*, \mathbf{h}(\hat{\mathbf{e}}_s, \mathbf{f}_s)\rangle\}_1^S)$
  repeat
    for $m = 1, \ldots, M$ **do**
      $\mathbf{d} = [0.0, \ldots, 0.0]$
      $d_m = 1.0$
      $\mathbf{w}'$ =Line-Search($\mathbf{d}, \mathbf{w}_x, \mathbf{Tr}$)
      $b\_scr' = \mathbf{BLEU}(\{\mathbf{r}_s, \mathrm{argmax}_{\hat{\mathbf{e}}_s \in \mathbf{C}_s} \langle \mathbf{w}', \mathbf{h}(\hat{\mathbf{e}}_s, \mathbf{f}_s)\rangle\}_1^S)$
      if $b\_scr' > b\_scr^*$ **then**
        $\mathbf{w}^* = \mathbf{w}'$
        $b\_scr^* = b\_scr'$
      end if
    end for
    $\mathbf{w}_x = \mathbf{w}^*$
  until no change in $\mathbf{w}^*$
  if $b\_scr^* > b\_scr$ **or** $x == 0$ **then**
    $\mathbf{w} = \mathbf{w}^*$
    $b\_scr = b\_scr^*$
end for
return $\frac{1}{\|\mathbf{w}\|}\mathbf{w}$

function Line-Search($\mathbf{d}, \mathbf{w}, \mathbf{Tr}$)
  $\mathbf{I} = \{\}$
  for $s = 1, \ldots, S$ **do**
    for all $\hat{\mathbf{e}}_s \in \mathbf{C}_s$ **do**
      $\hat{\mathbf{e}}_s.m = \langle \mathbf{d}, \mathbf{h}(\hat{\mathbf{e}}_s, \mathbf{f}_s)\rangle$ $//slope$
      $\hat{\mathbf{e}}_s.b = \langle \mathbf{w}, \mathbf{h}(\hat{\mathbf{e}}_s, \mathbf{f}_s)\rangle$ $//intercept$
    end for
    $i = 0$
    $best_i = \mathrm{argmin}_{\hat{\mathbf{e}}_s \in \mathbf{C}_s} \hat{\mathbf{e}}_s.m$ $//\hat{\mathbf{e}}_s.b\ breaks\ ties$
    $x_i = -\infty$
    repeat
      $i = i + 1$
      $best_i = \mathrm{argmin}_{\hat{\mathbf{e}}_s \in \mathbf{C}_s} \{\frac{best_i.b - \hat{\mathbf{e}}_s.b}{\hat{\mathbf{e}}_s.m - best_i.m}\} > x_{i-1}$
      $x_i = \frac{best_i.b - best_{i+1}.b}{best_{i+1}.m - best_i.m} > x_{i-1}$
    until No more intersection points found
    $add(\mathbf{I}, x_1^i)$
  end for
  $add(\mathbf{I}, max(\mathbf{I}) + \epsilon)$
  $x_{best} = \mathrm{argmax}_{x \in \mathbf{I}} \mathbf{BLEU}(\{\mathbf{r}_s, \mathrm{argmax}_{\hat{\mathbf{e}}_s \in \mathbf{C}_s} \langle \mathbf{w} + (x - \epsilon)\mathbf{d}, \mathbf{h}(\hat{\mathbf{e}}_s, \mathbf{f}_s)\rangle\}_1^S)$
  return $\mathbf{w} + (x_{best} - \epsilon)\mathbf{d}$

上の擬似コードでは入力パラメータとランダム生成した X 個のパラメータによって最適解を探索している．また，M 次元のパラメータに対して，1 次元ずつ line-search による最適化を行い，BLEU 値を最も大きくした次元だけを更新するという作業をパラメータの変化がなくなるまで繰り返している．

line-search では全ての訓練データに対する K-best の解集合で図 1 に示したような 1-best を規定する line を求めている．1-best を規定する line の求め方は最初に最も slope の低いものを求め (複数ある場合は intercept の高いものを選ぶ)，その直線に対して最も x 座標の小さい交点を持つ直線を選び，今度はその直線に対して前の交点よりも x 座標が大きい交点の中で最小の x 座標を持つ直線を選ぶという作業を交点が見つからなくなるまで繰り返すことで行う．これより，ある α の値 (x 座標の値) に対して，全ての訓練データにおける K-best において 1-best が求まるので，全ての 1-best から計算された BLEU の値が最大となるときの α を求めることができる．

実装ではこの Algoritm1 の作業後に更新されたパラメータで再度翻訳を行い，再び Algorithm1 を行っている．この作業をパラメータの変化がなくなるまで繰り返すことで最適化を行っている．

4 Minimum empirical risk Training(MerT)

構造的出力の予測問題を

$$F : \mathscr{X} \to \mathscr{Y}$$

とする．$\mathscr{X}$ は入力空間，$\mathscr{Y}$ は出力空間とする．また，訓練データの集合を

$$S = ((x_1, y_1), \cdots, (x_l, y_l)) \in (\mathscr{X} \times \mathscr{Y})^l$$

として表す．ここで確率分布 $P(x, y)$ の下で関数 F が誤識別する確率 (期待的リスク) は

$$R_{exp}(F) = \int_{\mathscr{X} \times \mathscr{Y}} \triangle(y, F(x)) dP(x, y)$$

となる．しかし通常，確率分布 $P(x, y)$ は未知であるため訓練データ S から求める経験的リスク

$$R_{emp}(F) = \frac{1}{l} \sum_{i=1}^{l} \triangle(y_i, F(x_i))$$

を学習問題では扱うこととなる．この経験的リスクを最小化するように学習する概念は経験的リスク最小化 (Empirical Risk Miniminzation) と呼ばれる．ここで訓練データを増加させたときに R_{emp} の最小化が R_{exp} の最小化と矛盾しない学習を考えると，Vapnik らによる構造的リスク最小化原理から Support Vector Machine(SVM) はこの条件を満たす学習法であると解釈され，R_{emp} に正則化項を考慮した形で定式化される．本稿ではこの SVM を SMT の最適化問題に適用することを考える．

従来，SVM は 2 クラス分類問題を扱う学習手法であったが，近年では多クラスの分類問題や構造的出力を扱うためにその一般化が盛んに行われている[8][9]．提案手法 MerT では Tsochantaridis らによって提案された多クラス分類のための Structural SVM[9][10] を基とする．

4.1 Structural Support Vector Machine

本稿第 3 節における定義と同様に訓練データを $\mathbf{T} = \{(\mathbf{f}_1, \mathbf{r}_1), \ldots, (\mathbf{f}_S, \mathbf{r}_S)\}$ とし，訓練データ $\mathbf{f}_s$ に対する K-best 出力の集合を $\mathbf{C}_s = \{\hat{\mathbf{e}}_{s,1}, \ldots, \hat{\mathbf{e}}_{s,K}\}$ として Structural SVM を基とした提案手法 MerT の定式化を行う．

4.1.1 ソフトマージン最適化

Structural SVM では正例と出力に対するスコアの差分 (マージン) を最大にするという一般化された最大マージン原理から L2 正則化項が導出される[9]．これは訓練データに対して線形分離可能であることを仮定して定式化されたものであり，ソフトマージン最適化では線形分離不可能な訓練データに対する誤りも考慮するため，非負の変数であるスラック変数 ξ_s を用いて次式のように定式化される．ただし，$\delta\mathbf{h}_s = \mathbf{h}(\mathbf{r}_s, \mathbf{f}_s) - \mathbf{h}(\hat{\mathbf{e}}_s, \mathbf{f}_s)$ とする．

$$\begin{aligned} &\min_{\mathbf{w},\xi\geq 0} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{S}\sum_{s=1}^{S}\xi_s \\ &s.t.\ \forall\hat{\mathbf{e}}_1 \in \mathbf{C}_1 \setminus \mathbf{r}_1 :\ \langle \mathbf{w}, \delta\mathbf{h}_1\rangle \geq 1 - \xi_1 \\ &\vdots \\ &s.t.\ \forall\hat{\mathbf{e}}_S \in \mathbf{C}_S \setminus \mathbf{r}_S :\ \langle \mathbf{w}, \delta\mathbf{h}_S\rangle \geq 1 - \xi_S \end{aligned} \tag{3}$$

式 (3) は L2 正則化項とスラック変数 ξ_s による経験的リスク R_{emp} を同時に最小化する問題となっており，λ は正則化とリスクの調節を行うパラメータとなる．

4.1.2 Margin-rescaling

Tsochantaridis らは式 (3) を正例と出力間の損失 $\triangle(\mathbf{r}_s, \hat{\mathbf{e}}_s)$ によってスケーリングすることを提案している[9][10]．これは損失 $\triangle(\mathbf{r}_s, \hat{\mathbf{e}}_s)$ の大きなサンプルに対してより大きなスラック変数の値を与えるためである．本稿では特にマージンに対してスケーリングを行う Margin-rescaling を用いる．

$$\begin{aligned} &\min_{\mathbf{w},\xi\geq 0} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{S}\sum_{s=1}^{S}\xi_s \\ &s.t.\ \forall\hat{\mathbf{e}}_1 \in \mathbf{C}_1 \setminus \mathbf{r}_1 :\ \langle \mathbf{w}, \delta\mathbf{h}_1\rangle \geq \triangle(\mathbf{r}_1, \hat{\mathbf{e}}_1) - \xi_1 \\ &\vdots \\ &s.t.\ \forall\hat{\mathbf{e}}_S \in \mathbf{C}_S \setminus \mathbf{r}_S :\ \langle \mathbf{w}, \delta\mathbf{h}_S\rangle \geq \triangle(\mathbf{r}_S, \hat{\mathbf{e}}_S) - \xi_S \end{aligned} \tag{4}$$

Margin-rescaling では式 (4) のように参照訳と翻訳文間のスコアのマージンである 1 を損失でスケーリングした形で定式化される．

ここで 1 つのサンプルに対する解集合の中で最も誤りを起こしている解におけるスラック変数の値は

$$\xi_s^* = \max\{0, \max_{\hat{\mathbf{e}}_s \in \mathbf{C}_s \setminus \mathbf{r}_s} \{\triangle(\mathbf{r}_s, \hat{\mathbf{e}}_s) - \langle \mathbf{w}, \delta\mathbf{h}_s\rangle\}\}$$

であり，制約の中で最も効果が期待できるものだけで最適化を行う場合，$\frac{1}{S}\sum_{s=1}^{S}\xi_s^*$ を最小化することとなる．ここで ξ_s^* は $\triangle(\mathbf{r}_s, \hat{\mathbf{e}}_s)$ に対して $\xi_s^* \geq \triangle(\mathbf{r}_s, \hat{\mathbf{e}}_s)$ であり，$\frac{1}{S}\sum_{s=1}^{S}\xi_s^*$ は $\frac{1}{S}\sum_{s=1}^{S}\triangle(\mathbf{r}_s, \hat{\mathbf{e}}_s)$ の上域をとることが証明されている[9]．

本稿では BLEU を用いて損失 $\triangle(\mathbf{r}_s, \hat{\mathbf{e}}_s)$ を次のように定義する．Q は BLEU による損失のスケールを行う定数とする．

$$\triangle(\mathbf{r}_s, \hat{\mathbf{e}}_s) = Q \times \{1.0 - BLEU(\mathbf{r}_s, \hat{\mathbf{e}}_s)\}$$

しかし，入力文と参照訳に対する素性のスコアを計算することは通常できないため，K-best の解集合 $\mathbf{C}_s$ の中から最も BLEU が高い解 $\hat{\mathbf{e}}_s^*$ を正解として 1 つ選ぶことでスコアの差分を

$$\delta\mathbf{h}_s = \mathbf{h}(\hat{\mathbf{e}}_s^*, \mathbf{f}_s) - \mathbf{h}(\hat{\mathbf{e}}_s, \mathbf{f}_s)$$

として計算する．同様に BLEU による損失を計算する場合にも $\hat{\mathbf{e}}_s^*$ を利用し

$$\triangle(\hat{\mathbf{e}}_s^*, \hat{\mathbf{e}}_s) = Q \times \{BLEU(\mathbf{r}_s, \hat{\mathbf{e}}_s^*) - BLEU(\mathbf{r}_s, \hat{\mathbf{e}}_s)\}$$

とする．これより式 (4) の最適化問題は

$$\begin{aligned} &\min_{\mathbf{w},\xi\geq 0} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{S}\sum_{s=1}^{S}\xi_s \\ &s.t.\ \forall\hat{\mathbf{e}}_1 \in \mathbf{C}_1 \setminus \mathbf{e}_1^* :\ \langle \mathbf{w}, \delta\mathbf{h}_1\rangle \geq \triangle(\mathbf{e}_1^*, \hat{\mathbf{e}}_1) - \xi_1 \\ &\vdots \\ &s.t.\ \forall\hat{\mathbf{e}}_S \in \mathbf{C}_S \setminus \mathbf{e}_S^* :\ \langle \mathbf{w}, \delta\mathbf{h}_S\rangle \geq \triangle(\mathbf{e}_S^*, \hat{\mathbf{e}}_S) - \xi_S \end{aligned} \tag{5}$$

となる．

4.1.3 最適化法

Structural SVM では一般に式 (5) の最適化問題の双対問題を考えて，SVM^{struct}[9]などを利用して最適化が行われる．しかし，本稿では目的関数を直接的に最小化できるという利点を持った MERT の line-search を用いて，式 (5) における主問題のままで最適化を行う．

MerT の最適化が MERT における最適化と異なる点は図 1 で表した直線の関数が

$$\left\{ \underbrace{\triangle(\hat{\mathbf{e}}_s^*, \hat{\mathbf{e}}_s) - \langle \mathbf{w}, \delta\mathbf{h}_s\rangle}_{intercept} + \alpha \underbrace{\langle \mathbf{d}, \delta\mathbf{h}_s\rangle}_{slope} \right\} \tag{6}$$

となる点である．また，SVM^{struct} では制約条件を全て考慮するのではなく，効果が最も期待できる制約だけを経験的リスクとして逐次的に追加していく Cutting-Plane-Algorithm[11]を用いて最適化問題の効率化を図るが，line-search による最適化では 1-best(MerT では式 (6) の max) を探索して目的関数を計算する作業でこの効率化を実現している．

4.2 1-Slack Formulation

式 (5) は S 個の訓練データに対しそれぞれスラック変数が定義された S-Slack Formulation となっている．この定義では sentence 単位の BLEU によって損失が計算されることになるが，2 節で前述したように通常，BLEU は corpus 単位で N-gram のカウント数を集計してから幾何平均を行う．そこでスラック変数を 1 つだけ定義する 1-Slack Formulation を用いて，corpus 単位の BLEU で目的関数が定義できるように拡張を行う．ただし，$\triangle(\{\hat{\mathbf{e}}_s^*, \hat{\mathbf{e}}_s\}_1^S) = Q \times \left\{\mathbf{BLEU}(\{\mathbf{r}_s, \hat{\mathbf{e}}_s^*\}_1^S) - \mathbf{BLEU}(\{\mathbf{r}_s, \hat{\mathbf{e}}_s\}_1^S)\right\}$ とする．

$$\min_{\mathbf{w},\xi\geq 0} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \xi \tag{7}$$

$$s.t.\ \forall(\hat{\mathbf{e}}_1, \cdots, \hat{\mathbf{e}}_S) \in \mathbf{C}^{\mathbf{S}} :\ \frac{1}{S}\sum_{s=1}^{S}\langle \mathbf{w}, \delta\mathbf{h}_s\rangle \geq \triangle(\{\hat{\mathbf{e}}_s^*, \hat{\mathbf{e}}_s\}_1^S) - \xi$$

式 (7) の学習基準は基本的には MERT の学習基準に L2 正則化項を考慮したものと解釈できる．ここで参照訳の代わりとして用いる $\mathbf{e}^*$ の選択方法は corpus 単位の最大 BLEU 値を Greedy 探索によって計算することで求める．

最適化においても式 (6) より sentence 単位の BLEU が入った形で直線が引かれることになるが，ここでは直線の関数を MERT と同様に式 (2) の Log-linear モデルで行う．また，1-Slack Formulation の最適化問題では corpus 単位の BLEU を用いているため，S-Slack Formulation とは異なり，SVM^{struct} などの従来法で実装することは容易ではない．

5 実験

5.1 実験条件

本実験では翻訳機に句ベースの統計的機械翻訳システム Moses[7]を使用した．また，翻訳モデルには GIZA++[12]，言語モデルには SRILM[13]を学習ツールとして使用した．Log-linear モデルの素性数は 14 で，4-gram 言語モデル，単語翻訳確率，句

翻訳確率，re-ordering(msd)，句ペナルティ，単語ペナルティから成る．

モデル学習用の対訳コーパスは IWSLT2008 で提供されたものを用いた．trainset に約 2 万文を使用し，重みパラメータの学習には devset4(約 500 文，7-references)，テストには devset5(約 500 文，7-references),devset6(約 500 文，6-references) を用いて実験を行った．devset5 は devset6 に比べて devset4 に近いデータである．

また，MerT のパラメータ λ は予備実験の結果から 0.01 とし，Q は予備実験から通して 10.0 で固定とした．重みパラメータ学習の際の K-best 出力数は 1000，ランダム生成する初期点のパラメータは 5 とした．

5.2 実験結果

評価尺度として corpus 単位の BLEU を用いて実験を行った．実験結果を表 1 に示す．

表 1 BLEU による翻訳評価

	devset5	devset6
MERT	19.58	24.92
MerT(S-Slack)	17.76(-△1.82)	22.92(-△2.00)
MerT(1-Slack)	18.83(-△0.75)	25.17(+△0.25)

表 1 から S-Slack Formulation による最適化では通常の MERT による最適化よりも翻訳精度が下がってしまっていることがわかる．これは S-Slack Formulation では sentence 単位の BLEU を平均化した形で最適化を行っていたが，評価実験においては corpus 単位での BLEU を用いて評価したことが原因であると考えられる．すなわち，最適化の際に設けられている目的関数と実際の評価実験で用いた尺度に違いがあったために最適化が適切に機能しなかったからであると考えられる．

次に 1-Slack Formulation による最適化について述べる．devset5 では MERT より翻訳精度が低くなったものの，学習データと乖離の大きい devset6 では MERT よりも高い翻訳精度が得られた．これは提案手法 MerT が MERT に比べて過学習を緩和する機能を持っていることを示唆している．

6 まとめ

本稿では Support Vector Machine に基づいた Log-linear モデルの学習法 MerT を提案した．これは BLEU の値を考慮した最大マージン原理に基づくもので，MERT の学習手法をそのまま活用することができる．

1-Slack Formulation による手法では基本的に MERT の目的関数に L2 正則化を加えたものとしてみることもでき，実験結果からは訓練データに対する過学習を緩和するような傾向が得られた．しかし，結果全体としては提案手法よりも MERT による実験結果の方がテストデータに対する BLEU 値を適切に上げることができていたと言える．

この原因として MERT では直接 BLEU 値を最大化しているのに対し，提案手法では目的関数への BLEU の寄与が少なかったことが挙げられる．これを解決する方法としては BLEU による損失をスケーリングする定数 Q を大きくすることが考えられる．また，Margin-rescaling ではなく，スラック変数自体をスケーリングする Slack-rescaling[9] [10]を用いることで BLEU 値による損失をより強調することも考えられる．

さらに本提案手法では正例の素性値が必要となるが，参照訳への素性値を計算することができないため，K-best 出力の中から参照訳を代用したことも問題と考えられる．MERT では参照訳と翻訳文の関係を直接目的関数としているのに対し，提案手法では参照訳と間接的な形で目的関数が定式化されるため，学習に問題が生じたと考えられる．

今後，上記した提案手法の問題点を改善し，Europarl[15]等のより大規模なコーパスを用いて実験を行っていく予定である．

謝辞

本研究は NTT コミュニケーション科学基礎研究所における実習活動の一環として行った．本実習の機会を与えて下さった同志社大学山本 誠一教授，片桐 滋教授，並びに NTT コミュニケーション科学基礎研究所 外村 佳伸所長に感謝致します．手法評価に当たっては，科学研究費補助金 (特定領域研究: 情報爆発 IT 基盤) の助成を受けた．

参 考 文 献

(1) Brown, P. F., Stephen, A., Della, P., Vincent, J., Della, P. and Robert, L. M.: The Mathematics of Statistical Machine Translation: Parameter Estimation, *Computational Linguistics*, Vol.19, No.2, pp.263-311, (1993).

(2) Och, F. J. and Ney, H.: Discriminative Training and Maximum Entropy Models for Statistical Machine Translation, *Proc. 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp.295-302, (July 2002).

(3) Och, F. J.: Minimum Error Rate Training in Statistical Machine Translation, *Proc. 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp.160-167, (July 2003).

(4) Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: BLEU:A method for automatic evaluation of machine translation, *Proc. 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp.311-318, (July 2002).

(5) Cortes, C. and Vapnik, V.: Support Vector Networks, *Machine Learning*, Vol.20, pp.273-297, (1995).

(6) Vapnik, V.: Statistical Learning Theory, *Wiley-Interscience*, (1998).

(7) Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A. and Herbst, E.: Moses: Open source toolkit for statistical machine translation, *Proc. 45th Demo and Poster Sessions of the Association for Computational Linguistics (ACL)*, pp.177-180, (June 2007).

(8) Joachims, T.: Learning to align sequences: A maximum-margin approach, on-line manuscript, (2003).

(9) Tsochantaridis, I., Joachims, T., Hofmann, T. and Altun, Y.: Support vector machine learning for interdepedent and structured output spaces, *Proc. International Conference on Machine Learning (ICML)*, pp.104-112, (2004).

(10) Tsochantaridis, I., Joachims, T., Hofmann, T. and Altun, Y.: Large margin methods for structured and interdepedent output variables, *Journal of Machine Learning Research (JMLR)*, Vol.6, pp.1453-1484, (2005).

(11) Joachims, T.: Training linear SVMs in linear time, *Proc. ACM SIGKDD International Conference On knowledge Discovery and Data Mining(KDD)*, pp.217-226, (2006).

(12) Och, F. J. and Ney, H.: A Systematic Comparison of Various Statistical Alignment Models, *Computational Linguistics*, Vol.29, No.1, pp.19-51, (2003).

(13) SRILM: http://www.speech.sri.com/projects/srilm

(14) Philipp, K.: Europarl: A parallel corpus for statistical machine translation, *Proc. MT-Summit*, (2005).

(15) Wolfgang, M., Franz, J. O., Ignacio, T., Jakob, U.: Lattice-based Minimum Error Rate Training for Statistical Machine Translation, *Proc. Empirical Methods in Natural Language Processing(EMNLP)*, pp.725-734, (2008).