原著論文 Original Paper

統計的機械翻訳のためのマージン最大化学習
機械翻訳精度向上に向けて

Max Margin Learning for Statistical Machine Translation
Toward Improvement of Machine Translation Accuracy

林 克彦 Hayashi Katsuhiko
同志社大学 情報工学研究科 Doshisha University
dti0708@mail4.doshisha.ac.jp

渡辺 太郎 Watanabe Taro
情報通信研究機構 National Institute of Information and Communication Technology
taro.watanabe@nict.co.jp

塚田 元 Tsukada Hajime
NTT コミュニケーション科学基礎研究所 NTT Comunication Science Laboratories
tsukada@cslab.kecl.ntt.co.jp

磯崎 秀樹 Isozaki Hideki
（同　上）
isozaki@cslab.kecl.ntt.co.jp

山本 誠一 Yamamoto Seiichi
同志社大学 情報工学研究科 Doshisha University
seyamamo@mail.doshisha.ac.jp

keywords: Support Vector Machines, Statistical Machine Translation, Minimum Error Rate Training, machine learning

Summary

Minimum error rate training (MERT) has been a widely used learning method for statistical machine translation to estimate the feature function weights of a linear model. MERT has an advantage to incorpolate an automatic translation evaluation metrics as BLEU scores to its objective function. Weight vector can directly be optimized with Line search algorithm using error surface on a given set of candidate translations. It efficiently searches the best parameter resulting the highest BLEU scores. In this paper, we presented a new training algorithm for statisitcal machine translation, inspired by MERT and Structural Support Vector Machines. We performed MERT optimization by maximizing the margin between the oracle and incorrect translations under the L2-norm prior. Our experimental results on Japanese-English speech translation task showed that BLEU scores obtained by our proposed method were much better than those obtained by MERT. We achieved the best improvement of BLEU about +3.0 over standard MERT.

1. まえがき

統計的機械翻訳 (Statistical Machine Translation) では異言語間の翻訳過程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，統計的機械翻訳では雑音のある通信路モデル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 [Brown 93]．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対数線形モデルが主流となっている [Och 02]．対数線形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．対数線形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．

対数線形モデルの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 [Och 02]，実際の評価基準を直接用いる識別学習法も提案され，最小誤り学習法 [Och 03a](MERT; Minimum error rate training) と呼ばれている．MERT では翻訳訳質自動評価尺度である BLEU[Papineni 01] や WER(Word error rate)[Niesen 00] などを直接目的関数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．また，MERT では最適化法として線形計画法に基づいた形式で最適化を行う Och 法 [Och 03a] が用いられており，少量のパラメータを最適化する際には勾配型の探索法よりも極めて有効な手法であることが知られている．一方で，MERT には過学習を緩和させる定式化が採用されていないため，過学習を頻繁に起こすという問題が指摘されている [Cer 08]．過学習をさけるために目的関数を平滑化する手法 [Och 03a] は検討されたが，BLEU 値に近似を入れる必要などがあり，有効な精度を示すことはできていない．

一方，機械学習分野では Structural Support vector machines(以下，Structural SVM) と呼ばれる多クラス分類問題や構造化出力問題を扱うためのマージン最大化に基づいた識別学習法が提案されている [Tsochantaridis 05, Joachims 09]．本稿では MERT における過学習の問題を緩和するため，Structural SVM の考えを定式化に採り入れることでマージン最大化に基づく MERT を提案する．提案法では訳質評価尺度 BLEU を近似することなく目的関数に導入でき，また Structural SVM が持つ正解サンプルと誤サンプルのモデルスコアを広げる学習効果も考慮される．Structural SVM には SVM^{struct} と呼ばれる最適化の枠組みが一般に利用されるが，少量パラメータから成る線形モデルを最適化する際には MERT で利用されている Och 法が極めて有効な手法となる．よって，提案法の最適化にも MERT と同様の最適化法を利用する．

以下，第 2 章では統計的機械翻訳における対数線形モデル，第 3 章では MERT の枠組みについて説明を行う．第 4 章では Structural SVM について概説し，第 5 章において MERT と Structural SVM を踏まえた上で提案手法の詳細を示す．第 6 章では MERT と提案手法による実験結果を示し，その考察を行う．第 7 章では関連研究について述べ，第 8 章ではまとめと課題を記す．

2. 対数線形モデルによる統計的機械翻訳

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

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \Pr(\mathbf{e}|\mathbf{f}) \tag{1}$$

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．従来，この条件付確率を雑音のある通信路モデルとしてベイズの定理から

$$\Pr(\mathbf{e}|\mathbf{f}) = \Pr(\mathbf{f}|\mathbf{e})\Pr(\mathbf{e}) \tag{2}$$

として翻訳モデル $Pr(\mathbf{f}|\mathbf{e})$，言語モデル $Pr(\mathbf{e})$ に分割してモデル化されてきた [Brown 93]．一方，近年ではこの条件付確率を

$$\begin{aligned}\Pr(\mathbf{e}|\mathbf{f}) &= p_{\mathbf{w}}(\mathbf{e}|\mathbf{f}) \\ &= \frac{\exp[\sum_{m=1}^{M} w_m h_m(\mathbf{e},\mathbf{f})]}{\sum_{e'} \exp[\sum_{m=1}^{M} w_m h_m(\mathbf{e}',\mathbf{f})]}\end{aligned} \tag{3}$$

として対数線形モデルでモデル化されることが一般的となっている [Och 02]．これは M 種類の素性関数 h を重み w で線形結合し，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．翻訳の探索過程 (デコーダ) 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．

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \langle \mathbf{w}, \mathbf{h}(\mathbf{e},\mathbf{f}) \rangle. \tag{4}$$

このモデルにおいて良い翻訳結果を得るためには線形結合重み w の最適化が必要である．

3. 最小誤り学習法

3·1 学習のための目的関数

MERT 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目的関数として BLEU のような評価尺度を用いることができる．一般に統計的機械翻訳では BLEU を目的関数として学習を行うため，以下では BLEU を用いて定式化を行う．

入力文 $\mathbf{f}$ とその参照訳 $\mathbf{r}$ から成る S 個の訓練データを $\mathbf{T} = \{(\mathbf{f}_1, \mathbf{r}_1), \ldots, (\mathbf{f}_S, \mathbf{r}_S)\}$ とし，入力文 $\mathbf{f}_s$ 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 の目的関数は

$$\underset{\mathbf{w}}{\operatorname{argmax}} \mathbf{BLEU}\left(\left\{\mathbf{r}_s, \underset{\hat{\mathbf{e}}_s \in \mathbf{C}_s}{\operatorname{argmax}} \langle \mathbf{w}, \mathbf{h}(\hat{\mathbf{e}}_s, \mathbf{f}_s) \rangle\right\}_1^S\right) \tag{5}$$

として表すことができる．BLEU は次式で示すような訳質評価尺度である．

$$BP \cdot \exp\left\{\frac{1}{N}\sum_{n=1}^{N} \log p_n(\{\mathbf{r}\}_1^S, \{\hat{\mathbf{e}}\}_1^S)\right\}. \tag{6}$$

$p_n(\cdot)$ は翻訳結果 $\{\mathbf{e}\}_1^S$ の参照訳 $\{\mathbf{r}\}_1^S$ に対する n-gram の重なりの適合率であり，BP は翻訳結果文が短い場合のペナルティである．また，1 文単位の BLEU は評価尺度として信頼性が低いため，通常，1 サンプルごとではなくサンプル集合全体で計算される尺度である．BLEU は割合として示す場合 0〜1 の値域を取り，数値が大きいほど良い翻訳結果であると言える．

3·2 最 適 化 法

MERT の最適化法には Och によって提案された手法が用いられる．Och 法では線形計画法に基づいて各翻訳結果に対する線形モデルを直線として定義した形で最適化問題が定義される．まず，M 次元ベクトル $\mathbf{d}$ を定義する．このベクトル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．Och 法ではこのベクトルを用いて最適化を行う次元の変化量 α を最適化する問題として考える．これらからモデルの定式化は

$$\begin{aligned}\hat{\mathbf{e}}_{s,best} &= \underset{\hat{\mathbf{e}}_s \in \mathbf{C}_s}{\operatorname{argmax}} \langle \mathbf{w} + \alpha\mathbf{d}, \mathbf{h}(\hat{\mathbf{e}}_s, \mathbf{f}_s) \rangle \\ &= \underset{\hat{\mathbf{e}}_s \in \mathbf{C}_s}{\operatorname{argmax}} \left\{\underbrace{\langle \mathbf{w}, \mathbf{h}(\hat{\mathbf{e}}_s, \mathbf{f}_s) \rangle}_{\text{切片}} + \alpha \underbrace{\langle \mathbf{d}, \mathbf{h}(\hat{\mathbf{e}}_s, \mathbf{f}_s) \rangle}_{\text{傾き}}\right\}\end{aligned} \tag{7}$$

と置き換えることができる．この式からは最適化を行う次元の値が α によって変わり，その結果，K-best 出力の中からモデルによって選ばれる 1-best が変わることがわかる．また，横軸を α，縦軸をモデルによる尤度 ($Score$) とした二次平面において，各 K-best 出力は式 (7) に示した傾きと切片から直線として表現することができる．これを図 1 に示す． 図 1 では 1 つのサンプルに対する 6-

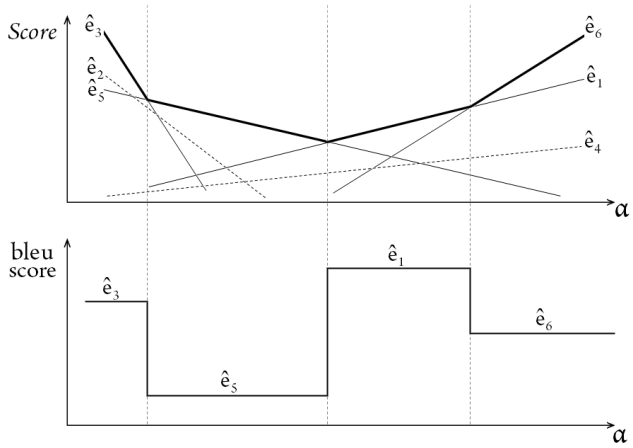

図 1 6-best による BLEU 値の局面

best の翻訳結果を例として示しているが，実際は MERT に使う全サンプルで図のような直線を定義する．図 1 の上図では交点を変わり目として選ばれる翻訳結果が変わること，下図ではその翻訳結果が選ばれたときの BLEU 値を示している．これより交点を境目として α を変えていくことで，最も BLEU 値が高くなる翻訳結果が選ばれるときの α を効率良く求めることができる (Line search algorithm).

Och 法の詳細は擬似コード 1,2 に示す．擬似コード 1 の 5-6 行目では各解に対して傾きと切片を求めている．8-17 行目では 1-best の変わり目となる交点を求めている．20-21 行目では交点から少量の値 ϵ ずらした位置 (実装上は交点間の中点) で BLEU 値を計算し，最大の BLEU 値を取る位置を探索している．擬似コード 2 では局所解に陥ることを避けるため，ランダムに生成した X 個の初期値から解の探索を行っている．8-24 行目では 1 次元ずつ最適化を行っていき，M 個の次元の中で最も良い BLEU 値が得られたものを次の重みとしている．この作業を重みの変化がなくなるまで繰り返し，さらに $X+1$ 個の重みベクトルの中で最も BLEU 値が大きな重みベクトルを正規化して最終的なパラメータとしている．実装上ではここからまた翻訳を行い，出力した翻訳結果をもとに再度 MERT を行っている．この作業もまた重みの変化がなくなるまで繰り返している．

4. Structural Support Vector Machines

近年，2 クラス分類問題を解くための SVM は多クラス分類問題や Structured Output Problem を扱うための一般化が行われてきた [Tsochantaridis 05, Joachims 09]．この SVM は Structural SVM と呼ばれており，正例と負例のスコアの差を拡げるという一般化されたマージン最大原理から L2 正則化項 (あるいは L1 正則化項) が導出される．さらにデータの誤差を許すため，スラック変数 ξ_s を導入したソフトマージンによる定式化が導かれ，次式

擬似コード 1 Line-optimize

1: $\mathbf{Input}: \mathbf{w}, \mathbf{d}, \mathbf{C}_1^S, \{\mathbf{r}_s, \mathbf{f}_s\}_1^S$
2: $\mathscr{I} = \{\}$
3: **for** $s = 1 \ldots S$ **do**
4: **for all** $\hat{\mathbf{e}}_s \in \mathbf{C}_s$ **do**
5: $\hat{\mathbf{e}}_s.m = \langle \mathbf{d}, \mathbf{h}_s(\hat{\mathbf{e}}_s, \mathbf{f}_s) \rangle$
6: $\hat{\mathbf{e}}_s.b = \langle \mathbf{w}, \mathbf{h}_s(\hat{\mathbf{e}}_s, \mathbf{f}_s) \rangle$
7: **end for**
8: $i = 0$
9: $l_i = \mathrm{argmin}_{\hat{\mathbf{e}}_s \in \mathbf{C}_s} \hat{\mathbf{e}}_s.m$
10: $x_i = -\infty$
11: **repeat**
12: $i = i + 1$
13: **if** $\{\frac{l_{i-1}.b - l_i.b}{l_i.m - l_{i-1}.m}\} > x_{i-1}$ **then**
14: $l_i = \mathrm{argmin}_{\hat{\mathbf{e}}_s \in \mathbf{C}_s} \{\frac{l_{i-1}.b - \hat{\mathbf{e}}_s.b}{\hat{\mathbf{e}}_s.m - l_{i-1}.m}\}$
15: $x_i = \{\frac{l_{i-1}.b - l_i.b}{l_i.m - l_{i-1}.m}\}$
16: **end if**
17: **until** no more intersections
18: $add(\mathscr{I}, x_1^i)$
19: **end for**
20: $add(\mathscr{I}, max(\mathscr{I}) + 2\epsilon)$
21: $x_{best} = \mathrm{argmax}_{x \in \mathscr{I}} \mathbf{BLEU}(\{\mathbf{r}_s, \mathrm{argmax}_{\hat{\mathbf{e}}_s \in \mathbf{C}_s} \langle \mathbf{w} + (x - \epsilon)\mathbf{d}), \mathbf{h}(\hat{\mathbf{e}}_s, \mathbf{f}_s) \rangle\}_1^S)$
22: **return** $\mathbf{w} + (x_{best} - \epsilon)\mathbf{d}$

の定式化となる．

$$\mathrm{argmin}_{\mathbf{w}, \xi \geq 0} \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{S} \sum_{s=1}^{S} \xi_s \tag{8}$$
$$s.t.\ \forall \hat{\mathbf{e}}_1 \in \mathbf{C}_1 \setminus \mathbf{r}_1 : \ \langle \mathbf{w}, \delta \mathbf{h}_1 \rangle \geq 1 - \xi_1$$
$$\vdots$$
$$s.t.\ \forall \hat{\mathbf{e}}_S \in \mathbf{C}_S \setminus \mathbf{r}_S : \ \langle \mathbf{w}, \delta \mathbf{h}_S \rangle \geq 1 - \xi_S.$$

ここで $\delta \mathbf{h}_s$ は正解の素性値と翻訳結果の素性値との差分であり

$$\delta \mathbf{h}_s = \mathbf{h}_s(\mathbf{r}_s, \mathbf{f}_s) - \mathbf{h}_s(\hat{\mathbf{e}}_s, \mathbf{f}_s) \tag{9}$$

となる．式 (8) における各制約条件は正解に対するモデルスコアの方が誤出力に対するモデルスコアよりも 1 以上大きくなければならないということを示している．このモデルでは制約条件が $O(S|C|)$ 個となり，一般には非常に大きな数となる．よって，この制約条件を全て考慮するのではなく，制約条件を破る危険性の高い誤出力を各サンプルから 1 つずつ選びながら，最適化することを行う．これは Cutting plane algorithm と呼ばれている．

4·1 Margin Rescaling

通常の Structural SVM は様々な方法により，現実問題により即した改良がなされている [Tsochantaridis 05,

擬似コード 2 複数の初期パラメータによる探索

```
Input :initial parameter w_0
Input : C_1^S, {r_s, f_s}_1^S
generate X random parameters w_1 ... w_X
b_scr = −∞
for x = 0 ... X do
    w* = w_x
    b_scr* = BLEU({r_s, argmax_{ê_s ∈ C_s} ⟨w*, h(ê_s, f_s)⟩}_1^S)
    repeat
        for m = 1 ... M do
            d = [0.0, ..., 0.0]
            d_m = 1.0
            w' = Line-optimize(w_x, d, C_1^S, {r_s, f_s}_1^S)
            b_scr' = BLEU({r_s, argmax_{ê_s ∈ C_s} ⟨w', h(ê_s, f_s)⟩}_1^S)
            if b_scr' > b_scr* then
                w* = w'
                b_scr* = b_scr'
            end if
        end for
        w_x = w*
    until no change in w*
    if b_scr* > b_scr then
        w = w*
        b_scr = b_scr*
    end if
end for
return 1/||w|| w
```

Joachims 09]．[Tsochantaridis 05] では hinge 損失の値をスケーリングするための手法として“ Margin rescaling ”と“ Slack rescaling ”と呼ばれる 2 種類の手法が提案されている．本稿では Margin rescaling のみを利用するので，ここでは Margin rescaling について説明を行う．Margin rescaling では式 (8) の制約条件が次式のように表される．

$$s.t.\ \forall \hat{\mathbf{e}}_s \in \mathbf{C}_s \setminus \mathbf{r}_s:\ \ \langle \mathbf{w}, \delta \mathbf{h}_s \rangle \geq \triangle(\mathbf{r}_s, \hat{\mathbf{e}}_s) - \xi_s.$$

ここでは損失 $\Delta(\mathbf{r}, \hat{\mathbf{e}})$ が大きなサンプル $\hat{\mathbf{e}} \neq \mathbf{r}$ に対して，損失が小さなサンプルよりも大きなマージンを設けるというペナルティをかしている．

4·2 1-Slack Formulation

[Joachims 09] では Cutting plane algorithm によって逐次的に最適化してゆく S-slack モデルに代わり，S 個のサンプルをまとめて最適化する 1-slack モデルを提案している．1-slack モデルではサンプル全体に対してひとつのスラック変数 ξ を持ち，最適化問題は次式のようになる．

$$\underset{\mathbf{w}, \xi \geq 0}{\operatorname{argmin}} \frac{\lambda}{2} \|\mathbf{w}\|^2 + \xi \tag{10}$$

$$s.t.\ \forall (\hat{\mathbf{e}}_1, \cdots, \hat{\mathbf{e}}_S) \in \mathbf{C}^S:$$

$$\frac{1}{S} \sum_{s=1}^{S} \langle \mathbf{w}, \delta \mathbf{h}_s \rangle \geq \frac{1}{S} \sum_{s=1}^{S} \Delta(\mathbf{r}_s, \hat{\mathbf{e}}_s) - \xi.$$

S-slack モデルと 1-slack モデルは本質的に等価であることが証明されており，2 次最適化問題において 1-slack モデルは S-slack モデルよりも効率良く最適化が行える [Joachims 09].

5. 提　案　法

ここでは Structural SVM の定式化を採り入れることでマージン最大化を考慮した MERT を提案する．また，1-slack モデルの定式化を用いることで，MERT と同様にサンプル集合単位での BLEU 値を考慮する．さらに最適化法には少量パラメータを効率的に最適化するため，MERT で用いられている Och 法を用いる．

$\delta \mathbf{h}_s$ の計算のためには，参照訳に対する素性値を求める必要がある．しかし，一般的な統計的機械翻訳方式では参照訳に対する素性値を求めることができないという問題がある．そこで K-best リスト中から参照訳に一番近い翻訳結果を参照訳とみなすこととする．以下ではまず参照訳の代替となる翻訳結果の選択方法について述べ，その後，定式化に必要な事項について述べる．

5·1 参照訳の代替となる翻訳結果の選択方法

[Venugopal 05] では K-best 出力から参照訳の代わりとなる翻訳結果を選択する手法が提案されており，本稿でもこの手法を用いる．BLEU 値は前述したように 1 文ごとで計算すると信頼性が低いため，1 文ごとに最も BLEU 値が高くなる翻訳結果を選ぶことは難しい．よって，[Venugopal 05] では 1 文ごとにその K-best リストから適当な初期翻訳結果を選択しておき，この初期値から Greedy 探索法によって BLEU 値が高くなる翻訳結果をそれぞれ更新していく手法を提案している．この作業を BLEU 値がそれ以上変化しなくなるまで繰り返し，その時点での翻訳結果集合をそれぞれの参照訳の代わりとして用いる．参照訳の代わりとなる翻訳結果を $\hat{\mathbf{e}}_s^*$ で表す．

5·2 Rescaling のための損失関数

ここでは式 (10) における Margin rescaling のための損失 Δ を BLEU によって定義することを考える．BLEU は 1 文ごとに計算することは難しいが，1-slack モデルでは $\frac{1}{S}\sum_{s=1}^{S} \Delta(\mathbf{r}_s, \mathbf{e}_s)$ として定義されるため，全文での BLEU 損失 $\Delta(\{\mathbf{r}_s, \mathbf{e}_s\}_1^S)$ によって近似することができる．ここで $\Delta(\{\mathbf{r}_s, \mathbf{e}_s\}_1^S)$ は

$$Q \times \{\mathbf{BLEU}(\{\mathbf{r}_s, \hat{\mathbf{e}}_s^*\}_1^S) - \mathbf{BLEU}(\{\mathbf{r}_s, \hat{\mathbf{e}}_s\}_1^S)\} \tag{11}$$

と定義する．Q は損失をスケールするための定数とし，モデルのハイパーパラメータである．これより 1-slack モデルでの最適化問題は次式のようになる．

$$\underset{\mathbf{w},\xi\geq 0}{\operatorname{argmin}}\frac{\lambda}{2}\|\mathbf{w}\|^2+\xi \tag{12}$$
$$s.t.\ \forall(\hat{\mathbf{e}}_1,\cdots,\hat{\mathbf{e}}_S)\in\mathbf{C}^S:$$
$$\frac{1}{S}\sum_{s=1}^{S}\langle\mathbf{w},\delta\mathbf{h}_s\rangle\geq\Delta(\{\hat{\mathbf{e}}_s^*,\hat{\mathbf{e}}_s\}_1^S)-\xi.$$

5·3 最適化法

式 (12) より提案法における目的関数は

$$\frac{\lambda}{2}\|\mathbf{w}\|^2-\frac{1}{S}\sum_{s=1}^{S}\langle\mathbf{w},\delta\mathbf{h}_s\rangle+\Delta(\{\hat{\mathbf{e}}_s^*,\hat{\mathbf{e}}_s\}_1^S) \tag{13}$$

として表すことができ，最適化法には Och 法を適用する．これは擬似コード 1 で示したアルゴリズムの 21 行目における目的関数を式 (13) に変えることで提案法にもそのまま適用することができる．しかし，目的関数を計算するためにはそれぞれのサンプルごとに $\hat{\mathbf{e}}_s$ を $(K-1)$-best の中から選び出す必要がある．本稿では $\hat{\mathbf{e}}_s$ の選択に擬似コード 1 で示した Line search algorithm からモデルスコアが最も高くなるものを選び出した．厳密には Cutting plane algorithm でも行われているように最も制約条件を犯しているサンプルを選択することが正しい定義となり，切片と傾きの計算は式 (7) ではなく次式で行うことが望ましい．

$$\underset{\hat{\mathbf{e}}_s\in\mathbf{C}_s}{\operatorname{argmax}}\left\{\underbrace{\Delta(\mathbf{e}_s^*,\mathbf{e}_s)-\langle\mathbf{w},\delta\mathbf{h}_s\rangle}_{\text{切片}}+\alpha\underbrace{\langle\mathbf{d},\delta\mathbf{h}_s\rangle}_{\text{傾き}}\right\}.$$

しかし，この計算では 1 文単位での近似された BLEU を計算する必要があるため，本稿の実験には利用しなかった．

6. 実験

6·1 実験システム

実験には一般的な句ベース統計的機械翻訳システム Moses[Koehn 07] を利用した．実験では Moses を用いて，パラメータチューニングのために 1000-best，翻訳評価のために 1-best の翻訳結果を出力した．Moses で使用した対数線形モデルの素性は以下のものである．

- N-gram 言語モデル $Pr(\mathbf{e})$: 目的言語の文章らしさを評価する．
- 単語翻訳モデル $t(e_i|f_j)$, $t(f_j|e_i)$: 原言語単語と目的言語単語の翻訳確率 [Brown 93]
- 句翻訳モデル $\phi(\overline{\mathbf{e}}|\overline{\mathbf{f}})$, $\phi(\overline{\mathbf{f}}|\overline{\mathbf{e}})$: 原言語句と目的言語句の翻訳確率
- 句並びかえモデル $p(m|\overline{\mathbf{f}},\overline{\mathbf{e}})$, $p(s|\overline{\mathbf{f}},\overline{\mathbf{e}})$, $p(d|\overline{\mathbf{f}},\overline{\mathbf{e}})$: 原言語と目的言語間におけるフレーズの位置関係を表す確率 [Koehn 05]
- 単語，句ペナルティ: 単語数やフレーズ数によって目的言語文の長さを制御する．

本実験では GIZA++[Och 03b] によって学習した翻訳モデルによって句辞書を抽出する [Koehn 07]．言語モデルの学習には SRILM toolkit[Stolcke 02] を使用し，4-gram 言語モデルを学習した．また，MERT と提案法におけるランダムな初期値は共に 10 個とした．

表 1 実験データ

データ		参照訳数	文章数	単語数
Training	ja	–	74910	1091525
	en	1	74910	941431
Development	ja	–	500	4133
	en	16	500	4212
Test1	ja	–	506	4227
	en	16	506	4265
Test2	ja	–	506	4454
	en	16	506	4235
Test3	ja	–	489	7082
	en	7	489	6756

6·2 実験データ

本稿では日英翻訳を対象として実験を行った．モデル学習用データ (Training data) には ATR 音声会話データ 7 万 5000 文，対数線形モデルの学習用開発データ (Development data) には IWSLT06 データ 500 文 (参照訳各 16 文) を用いた．テストデータ (Test data) には IWSLT06 データから 3 セット用意した．Test data1 は 500 文 (参照訳各 16 文)，Test data2 は 500 文 (参照訳各 16 文)，Test data3 は 489 文 (参照訳各 7 文) である．各データの詳細は表 1 に示す通りである．

6·3 ハイパーパラメータの設定

提案法では式 (11) と (13) に示したように Q と λ という 2 種類のハイパーパラメータが存在する．λ は正則化項と期待損失の調節を行う，Q は BLEU 損失の値をスケールするための変数である．図 2 は $Q=1000$ に固定した状態で λ を変化させたときの目的関数を示したものである (横軸は言語モデルに対する重み係数，縦軸は目的関数の値を示している)．ここからは Q を大きく考慮するために λ を小さくすると目的関数は MERT に近づき，逆に λ を大きくすると正則化項の支配が強くなることがわかる．

本稿ではハイパーパラメータの調節にクロスバリデーション法を用いた．データにはチューニングデータを 2 分割したものを使った．クロスバリデーション法による翻訳結果の BLEU 値の平均を表 2 に示す．表 2 の結果からは $Q=1000$ としたときが最も安定した結果となった．よって，本実験でのテストは $Q=1000$ として行っ

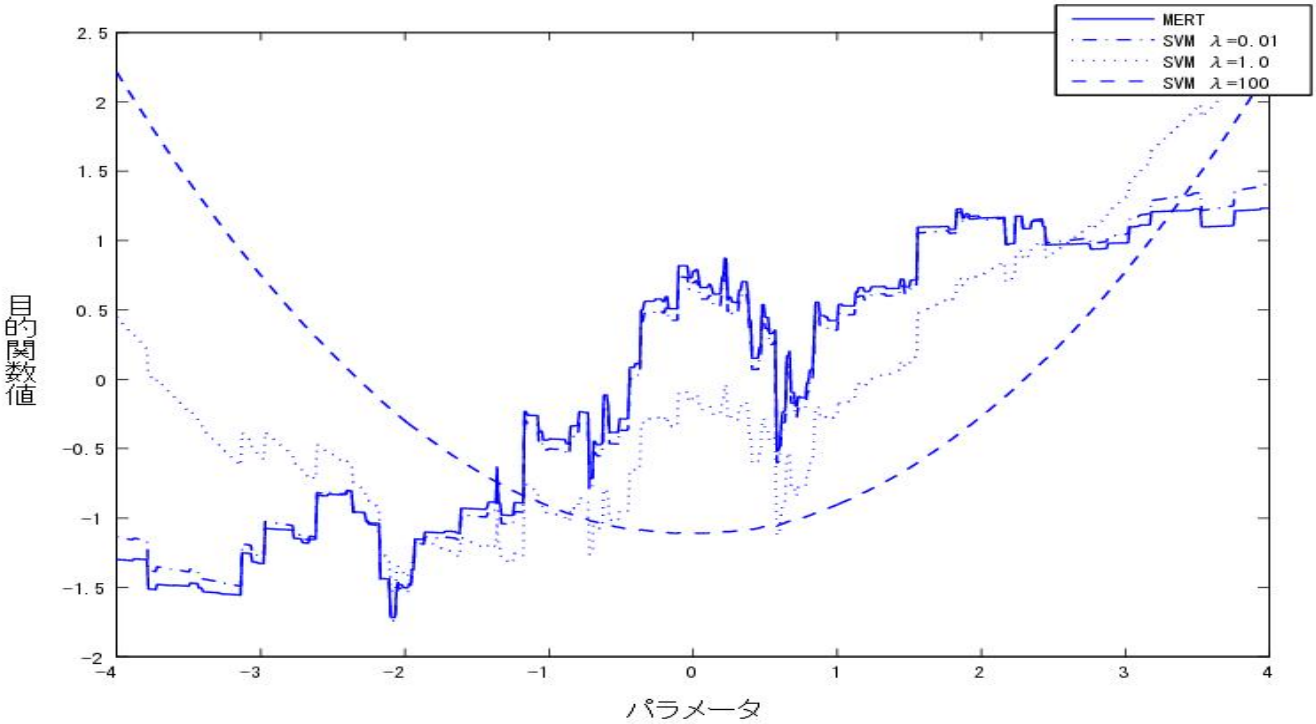

図 2 パラメータと目的関数の関係

表 2 クロスバリデーション法によるハイパーパラメータ設定

MERT	21.56	21.56	21.56	21.56	21.56
Hyper Parameter	$Q = 100000$	$Q = 10000$	$Q = 1000$	$Q = 100$	$Q = 10$
$\lambda = 1.0$	20.32	**23.48**	**22.44**	**22.21**	19.73
$\lambda = 0.1$	21.25	21.5	**22.835**	21.435	**23.15**
$\lambda = 0.01$	21.25	**21.645**	**21.61**	20.695	19.87
$\lambda = 0.001$	19.73	21.23	**23.635**	20.495	19.87

た．一方，$Q = 1000$ としたときの λ の値は良い結果を示した $\lambda = 0.001$ と $\lambda = 0.1$ を単純に採用した．

6·4 実　験　1

ここでは MERT と提案法の比較を行う．学習における収束条件等は全て同じ設定とした．実験ではそれぞれの手法で学習したモデルに対して 3 種類のテストデータを用いて翻訳実験を行った (評価値は BLEU とする)．その結果を表 3，4，5 に示す．実験結果からは条件の異なる複数のテストデータで，提案法の方が MERT に比べ，安定して高い BLEU 値を得られたことがわかる．また，学習用開発データのサイズを変化させた場合も同様に MERT と比較して安定した結果が得られた．ただし，学習用開発データのサイズが多くなるに従い (また極端に少ない場合)，両者の差は小さくなる傾向も見てとれる．

MERT と提案法における収束時間に関しては内部イテレーション (擬似コード 2 の 8-20 行目) の平均回数 (翻訳と学習を繰り返す外部イテレーション 1 回につき) の観点から議論を行う．ちなみに，本実験において外部イテレーションの回数は MERT と提案法でほぼ同回数 (4〜6 回) となった．学習用開発データのサイズが 500 文のとき，平均内部イテレーション 46 回，提案手法 ($Q = 1000$, $\lambda = 0.1$) では 605 回，提案手法 ($Q = 1000, \lambda = 0.001$) では 585 回となった．このことより提案手法は MERT に比べ，内部イテレーションの収束に 10 倍以上の時間がかかることがわかる．これは提案法では目的関数に直接パラメータを考慮する項があるため，パラメータの変化が大きくなり，収束に時間がかかっているからであると考えられる (実装上，内部イテレーションの収束条件は更新前と更新後のパラメータによる目的関数値が 0.00001 より下になるかどうかで判断している)．

6·5 実　験　2

実験 1 では提案法の方が MERT よりも安定して有効な手法であることが示せた．ここではその要因がどこにあるのかを調べるため，式 (12) で示した提案法の目的関数における要素を 3 つに分けることを考えた．1 つ目は BLEU 損失 $\Delta(\{\mathbf{r}_s, \mathbf{e}_s\}_1^S)$，2 つ目は正解 (の代替) と誤訳の差分 $\frac{1}{S}\sum_{s=1}^{S}\langle \mathbf{w}, \delta\mathbf{h}_s\rangle$，3 つ目は正則化項 $\frac{\lambda}{2}\|\mathbf{w}\|^2$ である．実験では 3 つの要素全て，差分と BLEU 損失，正則

表 3 Test data1 に対する実験結果

Method \ Dev Data Size	100	200	300	400	500
MERT	19.69	18.41	22.45	23.24	22.84
$Q = 1000$, $\lambda = 0.1$	**20.21**	**21.48**	**23.23**	**23.72**	**23.12**
$Q = 1000$, $\lambda = 0.001$	18.7	**20.13**	**24.45**	**23.8**	**24.82**

表 4 Test data2 に対する実験結果

Method \ Dev Data Size	100	200	300	400	500
MERT	29.09	26.58	29.06	31.14	31.5
$Q = 1000$, $\lambda = 0.1$	28.95	**29.64**	**31.42**	**32.96**	**32.15**
$Q = 1000$, $\lambda = 0.001$	**29.63**	**29.53**	**32.71**	**33.03**	**32.39**

化項と BLEU 損失，BLEU 損失，MERT(BLEU) を比較した．データは Development data500 文と 3 種類のテストデータを用いて実験を行い，その結果を表 6 に示す．

実験結果からは差分と BLEU 損失を考慮した場合が最も良い結果となった．これより提案手法が MERT よりも有効であった要因の 1 つとして正解と誤訳の差分を拡げるように学習するという項が非常に効果的であったということがわかる．また，正則化項は本実験においてそれほど有効な役割を果たしていないということが結果からはわかる．しかし，表 2 のハイパーパラメータの設定を行った実験からは正則化項の効果が見られる場合もあり，実用上は用いるデータに併せて適切なハイパーパラメータ調整を行うことが重要であると考えられる．

7. 関 連 研 究

従来，MERT の平滑化法としては [Och 03a, Cer 08] が提案されている．Och は MERT の提案と共に各翻訳結果の生成確率を用いた期待損失による関数の平滑化を提案している．これは目的関数の平滑化を行うという点で提案手法と共通しているが，提案手法では Och の平滑化法のように 1 文単位の BLEU 値を計算する必要がないため，提案法の方が安定した学習効果が期待できるといえる．Cer は Line search algorithm を行う際にその時点での最大 BLEU 値をみるのではなく，窓幅を持たせて近傍のパラメータにおける BLEU 値も考慮して最適化を行った．この手法は直接目的関数を平滑化するものではないため，提案手法と併用することも可能である．

MERT では局所解に陥ることを避けるため，ランダムに生成した複数の初期点を設けて最適化を行う．[Moore 08] ではメトロポリスサンプリング法を用いて初期点になるべく BLEU 値が良くなるパラメータを使うことで学習精度を向上させた．提案手法ではランダムに生成した初期点を用いているが，サンプリングで生成した初期値を提案法にも適用することでより高い学習効果を期待することができるであろう．

本稿での実験とは異なり，スパースで膨大な量の素性を用いたモデル [Watanabe 07, Chiang 08, Chiang 09] での学習法も提案されている．Watanabe は [Crammer 06] によって提案された Margin infused relaxed algorithm(MIRA) を高次元モデルの学習に適用した．Chiang も同様の実験を行っている．さらに [Liang 06] は Perceptron による学習法の有効性を示している．提案手法では少量パラメータを効率良く最適化するため Och 法を用いたが，SVM の 2 次最適化問題を解く SVM^{struct} の枠組みを用いれば，これらの手法同様に高次元モデルにも適用可能となろう．Perceptron では BLEU 値を考慮することができず，MIRA では 1 サンプルごとに BLEU 値を計算する必要があるため，提案手法の全サンプルで BLEU 値を計算できることは利点として挙げることができる．

8. む す び

本稿ではマージン最大化の定式化から導かれる目的関数を使うことで MERT に代わる新しい学習法を提案した．提案法では，マージン最大化の定式化に 1-slack Structural SVM を用い，全サンプル単位での BLEU 損失以外に，正解と誤訳のスコアの差分，正則化項を目的関数に考慮する．日英会話データの翻訳実験で，提案法は MERT に比べて極めて高い翻訳精度を達成した．本実験では正解と誤訳のスコアの差を拡げるという項を考慮することが翻訳精度向上において特に有効であることが確認された．一方で，実験結果からは正則化項の効果を確認することはできなかったが，用いるデータによっては正則化項も重要な役割を果たすため，実用上はハイパーパラメータを適切に設定することが重要であるといえる．

一方，提案法には正解として用意した参照訳に対して素性値を直接計算できないという問題点がある．これは提案法のみでなく，MIRA や Perceptron などでもそうであり，[Liang 06] では参照訳の素性値を計算する方法も提案されているが，現状では K-best から参照訳を代用することが最も有効な手法となっている．また，1 サンプ

表 5 Test data3 に対する実験結果

Method \ Dev Data Size	100	200	300	400	500
MERT	12.47	10.58	12.91	12.63	13.93
$Q = 1000$, $\lambda = 0.1$	11.36	**12.98**	**14.2**	**14.34**	**14.41**
$Q = 1000$, $\lambda = 0.001$	12.37	**12.9**	**14.58**	**14.41**	**14.46**

表 6 提案法の有効性における要因の検証

考慮する要素 \ Test Data	Test1	Test2	Test3
全て	23.12	32.15	14.41
差分, BLEU 損失	**24.82**	**32.39**	**14.46**
正則化項，BLEU 損失	22.96	31.0	13.26
BLEU 損失	23.27	31.92	13.47
MERT(BLEU)	22.84	31.5	13.93

ルごとの BLEU 値を上手く近似することで，Line search algorithm における 1-best の選択を目的関数の制約条件に則した形で行うことも今後の課題であると言える．1 サンプルごとの BLEU 値を計算する手法は提案されているが [Watanabe 06, Liang 06, Chiang 08]，精度の観点から現時点では実用的でない．

本稿では，比較的小規模な学習データを用いた日英翻訳実験で提案法の有効性を立証した．他の言語対やより大規模な学習データを用いた評価は，別途重要な課題だといえる．また，大規模データを用いる場合，学習の収束時間が遅いことは大きな問題である．学習時間を高速化する手法の研究も重要な課題となろう．

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と共同研究活動を通して行った．また，研究内容に対して的確なアドバイスを下さった NTT コミュニケーション科学基礎研究所，須藤克仁，鈴木潤，平尾努研究員，並びに，同志社大学，西田昌史准教授に深く感謝致します．

◇ 参 考 文 献 ◇

[Brown 93] Brown, P. F., Pietra, S. A. D., Pietra, V. D. J., and Mercer, R. L.: The mathematics of statistical machine translation: Parameter estimation, *Computational Linguistics*, Vol. 19, pp. 263–312 (1993)

[Cer 08] Cer, D., Jurafsky, D., and Manning, C. D.: Regularization and search for minimum error rate training, in *Proc. The 3rd Workshop on Statistical Machine Translation*, pp. 26–34 (2008)

[Chiang 05] Chiang, D.: A hierachical phrase-based model for statistical machine translation, in *Proc. the 43th Annual Meeting of the Association for Computational Linguistics*, pp. 263–270 (2005)

[Chiang 08] Chiang, D., Marton, Y., and Resnik, P.: Online Large-Marge Training of Syntactic and Structural Translation Features, in *Proc. Conference on Empirical Methods in Natural Language Processing*, pp. 224–233 (2008)

[Chiang 09] Chiang, D., Knight, K., and Wang, W.: 11,001 New Features for Statistical Machine Translation, in *Proc. the North America Chapter of the Association for Computational Linguistics* (2009)

[Crammer 06] Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y.: Online passive agressive algorithms, *Machine Learning*, Vol. 7, pp. 551–585 (2006)

[Freund 96] Freund, Y. and Schapire, R. E.: Experiments with a New Boosting Algorithm, in *Proc. of the 13th International Conference on Machine Learning*, pp. 148–156 (1996)

[Joachims 98] Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features, in *Proc. the 9th European Conference on Machine Learning* (1998)

[Joachims 09] Joachims, T., Finley, T., and Yu, C.-N.: Cutting-Plane Training of Structural SVMs, *Machine Learning*, Vol. 1, pp. 27–59 (2009)

[Koehn 05] Koehn, P., Axelrod, A., Mayne, A.-B., Callison-Burch, C., Osborne, M., and Talbot, D.: Edinburgh System Description for 2005 IWSLT Speech Translation Evaluation, in *Proc. the 2nd International Workshop on Speech Language Translation* (2005)

[Koehn 07] 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, in *Proc. the 45th Annual Meeting of the Association for Computational Linguistics*, pp. 177–180 (2007)

[Kudo 00] Kudo, T. and Matsumoto, Y.: Japanese Dependency Structure Analysis Based on Support Vector Machines, in *Proc. Joint Sigdad Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* (2000)

[Liang 06] Liang, P., Bouchard-Cote, A., Klein, D., and Taskar, B.: An end-to-end discriminative approach to machine translation, in *Proc. the 44th Annual Meeting of the Association for Computational Linguistics* (2006)

[McDonald 05] McDonald, R., Crammer, K., and Pereira, F.: Online large-margin training of dependency parsers, in *Proc. the 43th Annual Meeting of the Association for Computational Linguistics*, pp. 91–98 (2005)

[Moore 08] Moore, R. C. and Quirk, C.: Random restarts in minimum error rate training for statistical machine translation, in *Proc. the 22nd International Conference on Computational Linguistics*, pp. 585–592 (2008)

[Niesen 00] Niesen, S., Och, F., Leusch, G., and Ney, H.: An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research, in *Proc. the 2nd International Conference on Language Resources and Evaluation* (2000)

[Och 02] Och, F. J. and Ney, H.: Discriminative training and maximum entropy models for statistical machine translation, in *Proc. the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 295–302 (2002)

[Och 03a] Och, F. J.: Minimum error rate training in statistical ma-

chine translation, in *Proc. the 41th Annual Meeting of the Association for Computational Linguistics*, pp. 160–167 (2003)

[Och 03b] Och, F. and Ney, H.: A systematic comparison of various statistical alignment models, *Computational Linguistics*, Vol. 29, pp. 19–51 (2003)

[Papineni 01] Papineni, K. A., Roukos, S., Ward, T., and Zhu, W.-J.: Bleu: a method for automatic evaluation of machine translation, in *Proc. the 39th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318 (2001)

[Quirk 05] Quirk, C., Menezes, A., and Cherry, C.: Dependency Treelet Translation: Syntactically Informed Phrasal SMT, in *Proc. the 43th Annual Meeting of the Association for Computational Linguistics*, pp. 271–279 (2005)

[Stolcke 02] Stolcke, A.: SRILM An Extensible Language Modeling Toolkit, in *Proc. International Conference on Speech Language Processing* (2002)

[Tsochantaridis 05] Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y.: Large margin methods for structured and interdependent output variables, *Machine Learning*, Vol. 6, pp. 1453–1484 (2005)

[Venugopal 05] Venugopal, A. and Vogel, S.: Considerations in maximum mutual information and minimum classiffication error training for statistical machine translation, in *Proc. European Association for Machine Translation* (2005)

[Watanabe 06] Watanabe, T., Suzuki, J., Tsukada, H., and Isozaki, H.: NTT Statistical Machine Translation for IWSLT 2006, in *Proc. the 3nd International Workshop on Speech Language Translation* (2006)

[Watanabe 07] Watanabe, T., Suzuki, J., Tsukada, H., and Isozaki, H.: Online Large-Margin Training for Statistical Machine Translation, in *Proc. Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 764–773 (2007)

〔担当委員：河原 大輔〕

2010 年 3 月 6 日　受理

著　者　紹　介

林　　克彦

2008 年同志社大学工学部情報システムデザイン学科卒業．2010 年同大学院工学研究科情報工学専攻卒業．現在，奈良先端科学技術大学院大学情報研究科博士後期課程所属．統計的機械翻訳の研究に従事．言語処理学会学生会員．

渡辺　太郎（正会員）

1994 年京都大学工学部情報工学科卒業．1997 年京都大学工学研究科情報工学専攻修士課程修了．2000 年 Language and Information Technologies, School of Computer Science, Carnegie Mellon University, Master of Science 取得. 2004 年京都大学博士 (情報学). 現在、情報通信研究機構専攻研究員. 主に統計的機械翻訳の研究に従事.

塚田　　元（正会員）

昭和６２年東京工業大学理学部情報科学科卒業．平成元年同大学院理工学研究科修士課程修了．同年ＮＴＴヒューマンインタフェース研究所に入社．平成８年ＡＴＲ音声翻訳通信研究所．平成１０～１１年 AT&T Labs. Research 滞在研究員．現在，ＮＴＴコミュニケーション科学基礎研究所に勤務．入社以降，音声合成，音声認識の研究開発を経て，現在は統計的機械翻訳の研究に従事．情報処理学会，言語処理学会，電子情報通信学会，日本音響学会，ACL 各会員.

磯崎　秀樹（正会員）

1983 年東京大学工学部計数工学科卒業．1986 年同工学系大学院修士課程修了．同年 NTT 入社．現在，NTT コミュニケーション科学基礎研究所 主幹研究員、博士（工学）．論理的推論・自然言語処理の研究に従事．1990 年より１年間 Stanford 大学客員研究員．平成 15 年度情報処理学会論文賞・山下記念研究賞、言語処理学会第 12, 13 回年次大会最優秀発表賞、Coling-ACL 2006 AFNLP Meritorious Asian Paper Award．電子情報通信学会，情報処理学会，言語処理学会会員.

山本　誠一（正会員）

昭和２５年生．昭和４７年大阪大学工学部電子工学科卒業．唱和４９年同大学院基礎工学研究科修士課程終了．同年国際電信電話株式会社入社．ＡＴＲ音声言語コミュニケーション所長などを経て現在同志社大学理工学部教授．工博．この間，適応信号処理，音声合成，音声認識，音声翻訳の研究に従事．1981 年度電子情報通信学会学術奨励賞，日本音響学会第 3 回技術開発賞，第 5 回技術開発賞，電子情報通信学会システムソサイエティ論文賞，電気通信普及財団テレコム技術賞などを受賞．日本音響学会，情報処理学会，言語処理学会，IEEE Fellow，電子情報通信学会 Fellow.