Vol.2015-NL-222 No.3
2015/7/15

# 訓練データにより適合した統計翻訳最適化戦略

今村 賢治[1,a)]　隅田 英一郎[1]

**概要**：統計翻訳は，対訳コーパスからモデルを学習し，翻訳を行うにも関わらず，訓練文を，必ずしも対訳のとおりに翻訳できるとは限らない．
本稿では，この訓練訳不一致現象に対して，句に基づく翻訳方式を例にとり，まずクローズドテストのエラー分析を行う．本稿の実験では，対訳が完全に一致するものは，約 6 割しかなく，約 3 割はモデルエラーであることを確認した．
次にモデルエラーに焦点をあて，訓練セットと開発セット両方を使った最適化方式を提案する．提案法は，実数で構成された密な素性の他に，二値で構成された疎な素性を導入し，開発セットと訓練セットで異なる素性を最適化する．実験では，開発セットで密な素性，疎な素性両方を，訓練セットで疎な素性を最適化したとき，テストセットの BLEU スコアをほとんど変えずに，訓練セットの完全一致率が 61.2%から 75.5%に向上し，BLEU スコアも 10 ポイント程度向上させることができた．

**キーワード**：句に基づく統計翻訳，訓練訳の不一致，モデルエラー，疎な素性，最適化

## 1. はじめに

現状の統計翻訳は，訓練したにも関わらず，訓練文を翻訳すると，その参照訳とは異なる翻訳結果になることが多い．これを本稿では，訓練訳の不一致現象と呼ぶ．この特性があるために，翻訳器の開発者は，ユーザに対して「この文は確実に翻訳できる」という保証ができず，常に不安が付きまとうことになる．

理想的な翻訳器は，訓練した対訳文に関しては必ず訓練した翻訳結果になってほしい．この理想を実現するため，本稿では，訓練文を翻訳するクローズドテストを評価基準に設定し，高得点を取ることで，訓練訳の一致率を向上させることを目標にする．ただし，テストセットの翻訳品質は落とさないこととする．

この目標を実現する簡単な方法は，翻訳メモリーに対訳文全体を登録し，主処理に先立って変換してしまうことであるが，その文しか翻訳することができない．本稿では，最適化プロセスを見直すことで，訓練文翻訳結果の一致率，および BLEU スコアをを向上させる．本稿で対象とする翻訳器は，句に基づく翻訳方式（以下，PBSMT）[6], [7] である．

本稿の貢献は，(1) 訓練文の参照訳不一致現象に焦点を当てて，その内訳を分析したこと，(2) モデルエラー減少のための戦略を提案したことの 2 点である．以下，第 2 節で PBSMT の概要を説明した後，第 3 節でクローズドテストを行い，現状の機械翻訳器における訓練文の翻訳結果一致率の測定と，エラーの分析を行う．第 4 節では，エラーの中でもモデルエラーに焦点を当て，訓練セットと開発セットを併用した最適化方式を提案する．第 5 章では実験を通じて提案方式の特徴を議論し，第 6 章でまとめる．

[1] 国立研究開発法人　情報通信研究機構
National Institute of Information and Communications Technology
[a)] kenji.imamura@nict.go.jp

## 2. 句に基づく統計翻訳

まず，本稿で対象とする句に基づく統計翻訳の仕組みを簡単に説明する [5]．統計翻訳は，大きく分けて，訓練フェーズ，最適化フェーズ，テストフェーズから成り立っている（**図 1**）．

訓練フェーズでは，訓練セット（対訳文集合）から句の対応を算出して，フレーズテーブルなどの翻訳モデルを作成する．また，訓練セットの目的言語側などを用いて，言語モデルを作成する．

テストフェーズでは，デコーダを使用して翻訳を行う．PBSMT では，以下の処理を実行する．

- 原文でフレーズテーブルを参照して，フレーズラティス（探索グラフ）を作る．**図 2** は，フレーズラティスの模式図である．図中の $\bar{f}$ はフレーズの原言語側，$\bar{e}$ は目的言語側を表す．
- ラティスを探索しながらスコア計算を行い，最もデ

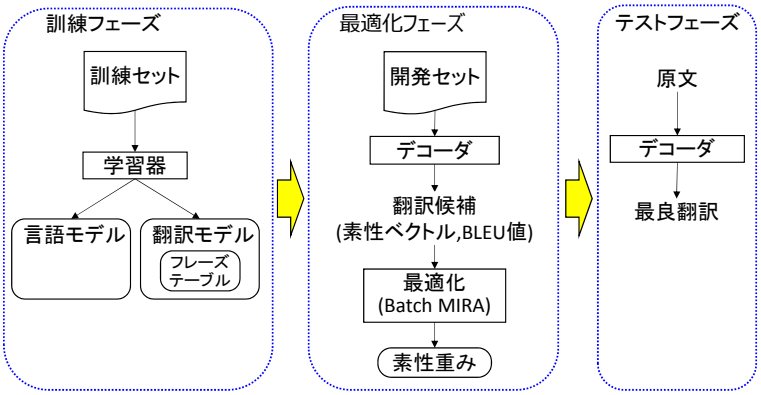



**図 1** 代表的統計翻訳の全体構成（訓練，最適化，テスト）
**Fig. 1** Structure of Major SMT (Training, Optimization, and Test)

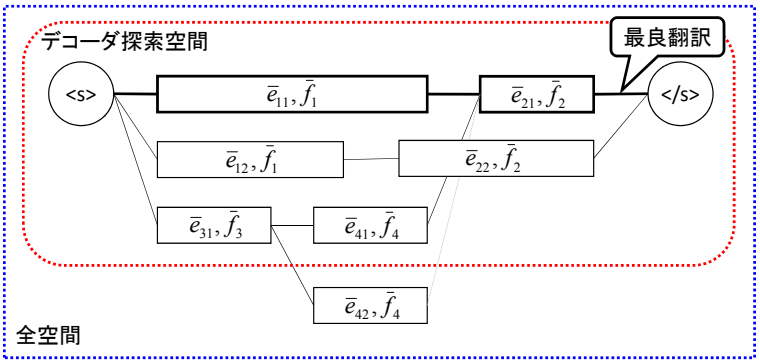



**図 2** PBSMT デコーダの探索空間
**Fig. 2** Search Space of Phrase-based Decoder

コーディングスコアが高いフレーズ列の目的言語側を，最良翻訳として出力する．探索には通常ビームサーチが用いられる．その場合，スコアが低い候補は探索中に枝刈りされ，探索空間から除かれる．

- デコーディングスコアは素性ベクトルと重みベクトルの内積（$\mathbf{w} \cdot \mathbf{h}(\mathbf{e}, \mathbf{f})$）で算出される．素性ベクトル $\mathbf{h}(\mathbf{e}, \mathbf{f})$ の要素には，翻訳モデルから得た句の翻訳確率（対数）や，言語モデルから算出した $n$ グラム対数確率など，十数種類が用いられることが多い．

最適化フェーズでは，素性の重みベクトル $\mathbf{w}$ を決定する．通常は，テストセットを模した対訳文（開発セット）を現在の重みで翻訳し，複数の翻訳候補と，各候補の素性ベクトル，および BLEU などの自動評価値を得る．最も自動評価値の高い候補が最良翻訳となるように，素性重みを調整する．

開発セットとテストフェーズで翻訳する文（テストセット）が類似したものであれば，最適化フェーズで調整した重みベクトルを使ってテストセットを翻訳すると，翻訳品質は向上する．しかし，その重みで訓練セットを翻訳した場合，訓練フェーズで対訳を学習していても，本来の参照訳から離れてしまう．

**表 1** コーパスサイズ
**Table 1** Corpus Size

| セット名 | 文数 | 備考 |
|---|---|---|
| 訓練 | 39,953 | 80 単語以下 |
| （クローズド） | 1,000 | 訓練セットから選択 |
| 開発 | 506 | |
| テスト（オープン） | 489 | |

## 3. クローズドテストにおけるエラー

本節では，現状の機械翻訳器における訓練文の翻訳精度を把握するため，クローズドテストを行う．まず，実験条件を述べるが，これは第 5 節で行う実験と共通する設定である．

### 3.1 実験設定

本稿では，IWSLT 2007 [2] の JE セット（分野は旅行会話）を使用し，英日翻訳を対象に実験を行う．コーパスサイズを**表 1** に示す．クローズドテスト用の文（クローズドセット）は，訓練セット全体から 1,000 文をランダムに選択し，精度測定に利用することとした．このコーパスは，機械翻訳としては小さいが，大規模コーパスも同じ傾向があると考え，実験しやすい本コーパスを利用した．

**表 2** クローズドテストにおける最良翻訳と参照訳との一致率とエラー分布（従来法）

**Table 2** Match Rates and Error Distribution in the Closed Test (Conventional Optimization)

| 分類 | 細分類 | 一致率 | （頻度） |
|---|---|---|---|
| **一致** | | 61.2% | (612) |
| | 1 | | (210) |
| | 2 | | (188) |
| | 3 | | (159) |
| | 4 | | ( 36) |
| | 5 以上 | | ( 19) |
| **モデルエラー** | | 31.9% | (319) |
| | タイプ 1 | | (261) |
| | タイプ 2 | | ( 58) |
| **探索エラー** | | 0.2% | ( 2) |
| **未知語** | | 6.7% | ( 67) |

詳細な実験設定は以下のとおりである．

- 前処理は，英語は PBSMT 翻訳ツール Moses[*1] の tokenizer と trucaser を使用して単語分割した．日本語は Unicode の NFKC 正規化をかけたのち，形態素解析器 MeCab[8] で単語分割した．
- 翻訳モデルは，Moses を使ってフレーズテーブル，語彙化並べ替えモデルを構築した．言語モデルは，KenLM[3] を使って日本語の 3-gram モデルを構築した．これらのモデルを対数線形結合して用いたが，重みは k-best Batch MIRA [1] を用いて，表 1 の開発セットに最適化した．
- デコーダは，エラー分析を探索グラフレベルで行うため，内部開発した Moses のクローンを用いた（設定方法はフレーズベースの Moses とほぼ同じ）．設定値は，ビーム幅= 200，`phrase_table_limit` = 20，`distortion_limit` は∞とした．

### 3.2 クローズドテスト結果の分析

まず，コーパスの各セットの BLEU スコア [10] は，クローズドセットが 79.05 ポイント，テストセットが 42.75 ポイントとなった（開発セットは 43.27 ポイント．いずれも 10 回最適化したときの平均値）．この数値からは，クローズドテストの BLEU スコアは，テストセットに比べると十分に高い．しかし，翻訳結果と参照訳が完全一致した場合，BLEU スコアが 100 ポイントになることを考えると，クローズドテストでも翻訳結果は参照訳に一致しない場合が多いことがわかる．

クローズドテストにおいて，最良翻訳と参照訳を比較した結果を**表 2** に示す．ここでは，比較結果を文献 [4] のエラー分類を参照し，以下のとおり分類した．

- **一致:** 最良翻訳が参照訳と一致．内訳に最良翻訳のフレーズ数を示す．フレーズ数 1 は，対訳文全体がフレーズテーブルに存在し，それが選択されたことを表す．
- **モデルエラー:** 最良翻訳のスコアより，参照訳スコアが低い．すなわち，モデルが訓練セットに最適化されていないことを表す．モデルエラーは，さらに以下の 2 タイプに分類した．
  - **タイプ 1:** 参照訳がデコーダの探索空間に存在．
  - **タイプ 2:** フレーズテーブルに存在するフレーズの組み合わせで参照訳を生成することができるが，`phrase_table_limit` などの制約により，デコーダの探索空間に参照訳が存在しない．
- **探索エラー:** 最良翻訳スコアより，参照訳スコアが低い．本来なら参照訳が最良翻訳として選ばれるべきだが，ビーム幅や `distortion_limit` 制約により探索中に枝刈りされた場合．
- **未知語:** 現在のフレーズテーブルでは，どう組み合わせても参照訳を作るのは不可能．文献 [4] では，このエラーをさらに原言語側が存在しないエラー (SEEN) と，原言語は存在するが，目的言語が存在しないエラー (SENSE) に詳細化したが，本稿では区別しなかった．

表 2 を見ると，最良翻訳と参照訳が一致したのは 61.2% と，決して高くはない数値となった．一致の内訳をみると，フレーズ数 1 は約 21%だったので，対訳全体をフレーズテーブルが保持していたわけではない．

モデルエラーは，全体の 31.9%であった．使用したモデルは参照訳を使って訓練したにも関わらず，参照訳自体のスコアが最高とはならなかった．

探索エラーはこの実験ではほとんど観察されなかったが，これは `distortion_limit` を $\infty$ にしたためである．ビーム幅は 200 で十分だった．

この実験では，未知語も約 7%存在した．この原因は詳細には分析していないが，訓練時に長すぎるフレーズが削除されるなどして，フレーズテーブルに含まれない場合があるためではないかと考えられる．

まとめると，訓練に使用した文を翻訳したにも関わらず，参照訳に一致するものはこの実験では 6 割程度しかなく，訓練セットにさらに適合できる余地を残している．

## 4. 提案方式

本稿では，上記エラータイプのうち，モデルの最適化が不十分であることに起因する，モデルエラータイプ 1 に焦点を絞る．言い換えると，フレーズテーブルの変更など，モデルそのものの変更は行わない．

本稿の目標は，訓練データ翻訳時に，参照訳との一致率を向上させることである．つまり，モデルを訓練データに適応させると言い換えることもできる．そのため，以下の 2 つの戦略を導入する．

[*1] `http://www.statmt.org/moses/`

**表 3** 素性テンプレート
**Table 3** Feature Templates

| 密な素性 (実数値) | | 疎な素性 (二値) | |
|---|---|---|---|
| 名前 | 備考 | 名前 | 備考 |
| 句の翻訳確率 | 原言語，目的言語双方向 | 句の共起 | $h(\bar{e}, \bar{f})$ |
| 句のモデル 1 確率 | 原言語，目的言語双方向 | | |
| 語彙化並び替え | (直前 \| 直後) × (順 \| 逆 \| 隣接せず) の 6 種 | 並び替えタイプ | $h(\bar{e}, \bar{f}, type)$．$type$ は 6 種のいずれか |
| distortion スコア | 隣接フレーズ距離 | | |
| 言語モデルスコア | $n$ グラム確率 | 1,2,3 グラム | $h(e_i)$, $h(e_i, e_{i-1})$, $h(e_i, e_{i-1}, e_{i-2})$ |
| 単語ペナルティ | 目的言語単語数 | | |
| フレーズペナルティ | フレーズ数 | | |
| 未知語ペナルティ | 未知語数 | | |
| | | 句の周辺単語 | $h(\bar{e}, \bar{f}, f_{i-1})$, $h(\bar{e}, \bar{f}, f_{i+1})$ |

(1) 最適化のためのチューニングセットに，訓練セットも使用する．ただし，テストセットにおける翻訳品質を下げないため，従来の開発セットも併用する（4.2 節参照）．
(2) 従来のスコア計算のための素性（本稿では密な素性と呼ぶ）は，たかだか十数種類しかなく，チューニングセットを増加させても調整できるパラメータは限られている．本稿では，疎な素性を追加導入し，チューニングによって最適化されるパラメータを大幅に増やす（4.1 節参照）．

### 4.1 疎な素性

従来の PBSMT では，翻訳スコア計算のための素性に，フレーズの翻訳確率（$\log Pr(\bar{e}|\bar{f})$）や，言語モデル確率（$\log Pr(e_i|e_{i-n+1}^{i-1})$）など，十数種類の実数値を使用してきた．これらは，ほぼすべての翻訳候補で 0 以外の数値が入ったベクトルになるため，密な素性 (dense feature) と呼ばれる．それに対し，引数が一致したときだけ 1，それ以外は 0 を返す二値関数を使った素性は，疎な素性 (sparse feature) と呼ばれる [11]．たとえば，以下は疎な素性関数の例である．

$$h(\bar{e}, \bar{f}) = \begin{cases} 1 & \bar{f} = \text{"バス"} \ \& \ \bar{e} = \text{"}bus\text{"} \text{のとき} \\ 0 & \text{それ以外} \end{cases}$$

本稿で使用した密な素性（のべ 15 種）と疎な素性の一覧を**表 3** に示す．疎な素性は，「句の周辺単語」を除き，密な素性を二値関数に変更したものである．疎な素性数は，使用するデータによって異なるが，後述する 5.2 節の実験では，9 万個程度である．

従来より，密な素性の重みパラメータを最適化する方法として，誤り率最小訓練法 (MERT[9]) がある．しかし，MERT は線分探索を素性ごとに繰り返すため，あまりに多次元の素性空間を最適化するには効率が悪いことが知られている．そのため，本稿では，Batch MIRA [1] を使用して素性重みの最適化を行う．Batch MIRA は，簡単に言うとオンラインマージン最大学習アルゴリズムを翻訳の最適化に適用したもの（たとえば [11]）の一種で，MERT の代替として機能する．また，MERT は密な素性しか最適化できないが，Batch MIRA は密な素性と疎な素性を区別せずに最適化できるという特徴がある．

### 4.2 訓練セットを用いた最適化

われわれの目標は，テストセットの翻訳品質を落とさずに訓練セットの翻訳品質を向上させることである．従来の方法は，開発セットで密な素性を最適化していた．これを訓練セットに適応させるために，密な素性を訓練セットで最適化してしまっては，テストセットの翻訳品質が落ちてしまう．

そこで今回は，データのセットによって最適化する素性を変更する．たとえば，開発セットを用いて密な素性を最適化し，訓練セットは疎な素性の最適化に使用する．素性の切り替えは，密な素性と疎な素性をつなげて一本の素性ベクトルにする際，最適化しない方の素性をゼロベクトルにする．そして，ベクトル全体を Batch MIRA に入力すればよい．5 節の実験では，どのセットでどのタイプの素性を最適化すべきか検証する．

なお，訓練セットは開発セットに比べてサイズが大きい．そのまま両方を使って最適化すると，訓練セットに過度にチューニングされてしまう恐れがある．この悪影響を低減するため，最適化時は開発セットをランダムにオーバーサンプリング（コピー）し，訓練セットと同数にする．

## 5. 実験

本節では，実験を通じて提案方式の特徴について議論する．特に断らない限り，実験設定は 3.1 節と同じである．

### 5.1 実験 1: 戦略別精度

本稿の提案方式は，訓練セット，開発セットそれぞれで，どの素性を最適化するかによって，方式にバリエーションを作ることができる．本節では，チューニングセットとし

**表 4** 組み合わせ別オープンテストの BLEU スコア
**Table 4** BLEU Scores of the Open Tests among Combinations

| 組み合わせ | | 訓練セット | | | |
|---|---|---|---|---|---|
| | | Dense | Sparse | Both | None |
| 開発セット | Dense | 41.14(-) | 44.06(+) | 42.16(-) | **42.75** |
| | Sparse | 38.61(-) | - | 39.57(-) | - |
| | Both | 39.77(-) | 42.59 | 41.63(-) | 44.54(+) |

**表 5** 組み合わせ別クローズドテストの BLEU スコア
**Table 5** BLEU Scores of the Closed Tests among Combinations

| 組み合わせ | | 訓練セット | | | |
|---|---|---|---|---|---|
| | | Dense | Sparse | Both | None |
| 開発セット | Dense | 82.00(+) | 84.36(+) | 88.78(+) | **79.05** |
| | Sparse | 82.62(+) | - | 87.48(+) | - |
| | Both | 82.46(+) | 88.92(+) | 89.01(+) | 76.56(-) |

て，開発セットとクローズドセット（1,000 文）を使用し，以下の組み合わせについて，翻訳品質を測定する．

- 開発セットは，密な素性 (Dense)，疎な素性 (Sparse)，両方 (Both) のどれかを最適化する．つまり，開発セットは必ずいずれかの素性を最適化する．
- クローズドセットは，密な素性，疎な素性，両方，なし (None) のどれかを最適化する．

開発セットで最適化する素性と訓練セットで最適化する素性の組み合わせを ⟨ 開発セットの素性, 訓練セットの素性 ⟩ と表現した場合，従来の最適化法は，⟨Dense, None⟩ という設定に相当する．

オープンテストの結果を**表 4** に，クローズドテストの結果を**表 5** に示す．数値はいずれも 10 回最適化を行ったときの平均値である．表中の (+) は，従来法 ⟨Dense, None⟩ に比べ，有意に向上したもの，(-) は有意に悪化したもの（t 検定危険率 5%）を表す．

オープンテストの品質は，⟨Both, None⟩ の組み合わせのとき，BLEU スコア最高値 (44.54) となった．従来法の設定 ⟨Dense, None⟩ に比べて有意に向上，または有意差なしだったのは，⟨Dense, Sparse⟩，⟨Both, Sparse⟩，⟨Both, None⟩ の 3 設定であった．

一方，クローズドテストの翻訳品質は，⟨Both, None⟩ 以外のすべての設定で従来法に比べ有意に向上した．オープンテストの精度が向上または変化なしと重なるのは，⟨Dense, Sparse⟩，⟨Both, Sparse⟩ の 2 設定である．つまり，開発セットは少なくとも密な素性を，訓練セットは疎な素性最適化した場合，テストセットの翻訳品質を下げずに訓練セットの品質を上げることができた．

### 5.2 実験 2: クローズドテストにおける一致率

提案方式における最良翻訳と参照訳との一致率を ⟨Both, Sparse⟩ の組み合わせで測定した．結果を，**図 3** の (b) に示す．なお，図 3(a) は，従来法の一致率で，表 2 をグラフ化したものである．

提案法による最適化の結果，⟨Both, Sparse⟩ の組み合わせでは，従来法に比べ，完全一致率は 61.2%から 75.5%に増加した．減少したエラーは，ほとんどがモデルエラーのタイプ 1 で，26.1%から 11.7%に減少した．その他のエラータイプは，提案法ではほとんど変化せず，それらを減少させるには，別の方法を必要とすることがわかった．

なお，オープンテストにおいても，同様に最良翻訳と参照訳の一致率およびエラータイプを調査したが，従来法と提案法でほとんど変化がなかった．

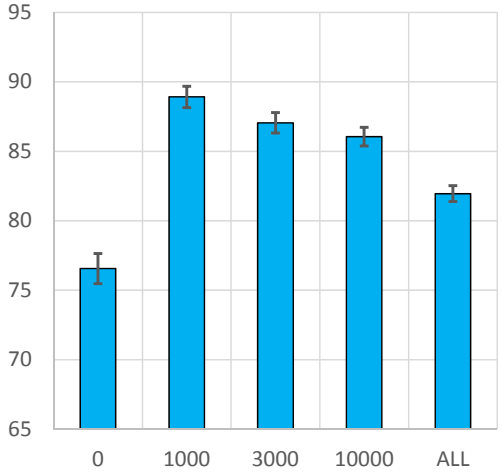



**図 4** 最適化用訓練セットサイズ別のクローズドテスト BLEU スコア
**Fig. 4** Closed Test BLEU Scores among Various Training Set Sizes for Optimization

### 5.3 実験 3: 最適化用訓練セットサイズ

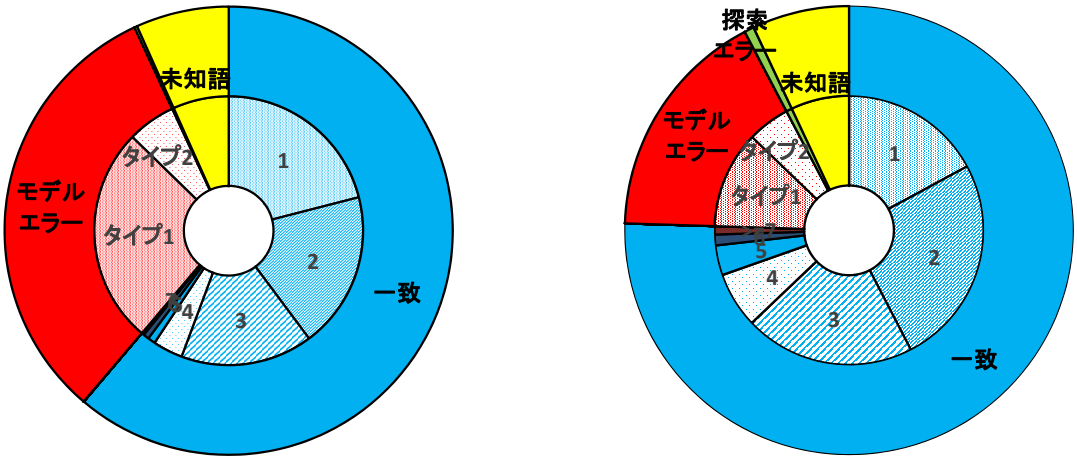
前節までは，最適化に使用した訓練セットは 1,000 文のみであったが，本節では，最適化に使用する訓練セットのサイズを変えて測定する．なお，素性の組み合わせは，⟨Both, Sparse⟩ である．クローズドテストの結果を**図 4** に示す．

グラフを見ると，最適化に使用する訓練セットは，小さいほどクローズドテストの翻訳品質がよく，サイズが大きくなるに従い，効果がほとんどなくなる．この原因はまだ判明していない．しかし，本稿の提案方式は，訓練セットが小さい場合に大きな効果を発揮することが確認された．

## 6. まとめ

本稿では，訓練訳の不一致現象について，その割合とエラータイプを分析した．本稿の実験設定では，参照訳と一致したのは 6 割強しかなく，3 割はモデルエラーであった．

さらに本稿では，このモデルエラーを減少させるため，密な素性の他に疎な素性を導入し，開発セット，訓練セットそれぞれで異なる素性を最適化した．その結果，開発




(a) 従来法による最適化　　(b) 提案法 ⟨Both, Sparse⟩ による最適化

**図 3**　クローズドテストにおける最良翻訳と参照訳の一致率

**Fig. 3**　Match Rates and Error Distribution in the Closed Tests

セットでは少なくとも密な素性，訓練セットでは疎な素性を最適化すると，テストセットの翻訳品質を低下させることなく，訓練セットの参照訳一致率を向上させることができた．実験では，一致率を 61.2%→75.5%まで上げることができた．

しかし，最適化に使う訓練セットのサイズを大きくすると効果なくなるという現象も判明したため，原因調査する必要がある．本稿では非常に小さなコーパスで実験したため，大規模データでも検証する必要がある．

訓練訳の不一致現象は，句に基づく統計翻訳に限らず，木構造を用いた統計翻訳にも存在すると考えられる．また，エラータイプに関しても，モデルエラーは存在すると考えられるため，他の方式における現象を確認したいと考えている．

## 参考文献

[1] Cherry, C. and Foster, G.: Batch Tuning Strategies for Statistical Machine Translation, *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Montréal, Canada, pp. 427–436 (online), available from ⟨http://www.aclweb.org/anthology/N12-1047⟩ (2012).

[2] Fordyce, C. S.: Overview of the IWSLT 2007 Evaluation Campaign, *Proceedings of the International Workshop on Spoken Language Translation 2007*, Trento, Italy (2007).

[3] Heafield, K., Pouzyrevsky, I., Clark, J. H. and Koehn, P.: Scalable Modified Kneser-Ney Language Model Estimation, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Sofia, Bulgaria, pp. 690–696 (online), available from ⟨http://www.aclweb.org/anthology/P13-2121⟩ (2013).

[4] Irvine, A., Morgan, J., Carpuat, M., Daumé III, H. and Munteanu, D.: Measuring Machine Translation Errors in New Domains, *Transactions of the Association ofr Computational Linguistics*, Vol. 1, pp. 429–440 (2013).

[5] Koehn, P.: *Statistical Machine Translation*, Cambridge University Press (2010).

[6] 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, *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, Prague, Czech Republic, pp. 177–180 (online), available from ⟨http://www.aclweb.org/anthology/P07-2045⟩ (2007).

[7] Koehn, P., Och, F. J. and Marcu, D.: Statistical Phrase-Based Translation, *HLT-NAACL 2003: Main Proceedings* (Hearst, M. and Ostendorf, M., eds.), Edmonton, Alberta, Canada, pp. 127–133 (2003).

[8] Kudo, T., Yamamoto, K. and Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis, *Proceedings of EMNLP 2004* (Lin, D. and Wu, D., eds.), Barcelona, Spain, pp. 230–237 (2004).

[9] Och, F. J.: Minimum Error Rate Training in Statistical Machine Translation, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (Hinrichs, E. and Roth, D., eds.), pp. 160–167 (online), available from ⟨http://www.aclweb.org/anthology/P03-1021.pdf⟩ (2003).

[10] Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: Bleu: a Method for Automatic Evaluation of Machine Translation, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318 (2002).

[11] Watanabe, T., Suzuki, J., Tsukada, H. and Isozaki, H.: Online Large-Margin Training for Statistical Machine Translation, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, pp. 764–773 (online), available from ⟨http://www.aclweb.org/anthology/D/D07/D07-1080⟩ (2007).

## 正誤表

**3.2 節（p.3，右段，13-14 行目）**

(誤)**探索エラー:** 最良翻訳スコアより，参照訳スコアが低い.

(正)**探索エラー:** 最良翻訳スコアより，参照訳スコアが高い.