

コーパス結合モデルと素性空間拡張に基づく統計翻訳のドメイン適応

Domain Adaptation Based on Corpus Concatenation and Feature Augmentation for Statistical Machine Translation

国立研究開発法人情報通信研究機構 先進的音声翻訳研究開発推進センター専門研究員 **今村 賢治**

2004年奈良先端科学技術大学院大学情報科学研究科情報処理学専攻博士課程修了。1985年日本電信電話株式会社。2014年株式会社ATR-TrekよりNICTに出向。機械翻訳の研究に従事。

国立研究開発法人情報通信研究機構 先進的音声翻訳研究開発推進センター 副センター長 **隅田 英一郎**

1999年京都大学大学院博士（工学）取得。1982年日本アイ・ビー・エム。2002年ATR。2010年NICT（2016年同フェロー）。機械翻訳の研究に従事。

1 はじめに

さまざまな種類のテキストが機械翻訳されるようになってきたが、すべての分野（ドメイン）において、適切に翻訳できる機械翻訳器はいまだに実現困難である。たとえば、特許文書を前提として作られた機械翻訳器で旅行会話を翻訳した場合、単語や言い回しが異なるため、多くの場合は不適切な翻訳になる。対象ドメインの翻訳品質を向上させる確実な方法は、学習データ（対訳文）を大量に収集し、翻訳器を訓練することであるが、さまざまなドメインについて、対訳文を大量に収集することはコスト的に困難なことが多い。ドメイン適応は、翻訳対象のドメイン（インドメイン）データが少なく、他のドメイン（アウトドメイン）データが大量にある場合、インとアウトのデータ双方を使って、インドメインの翻訳品質を向上させる技術である。我々は、これは機械翻訳を実用に供する際のキーとなる技術と考えている。

ドメイン適応は重要な技術であるがゆえに、さまざまな既存研究が存在する。本稿では、フレーズベース統計翻訳を前提として、まず既存のドメイン適応技術を概観する。次に、我々が提案するドメイン適応方式について述べる。我々が提案する方式は、複数ドメインを前提と

しており、複数のサブモデル（フレーズテーブル、言語モデル、語彙化並び替えモデルなど）を対数線形モデルで組み合わせる方法である。シンプルな方法であるが、機械学習分野のドメイン適応方法である素性空間拡張法^[1]の考え方を流用することで、各ドメインのいいとこどりをやっている。

2 統計翻訳のドメイン適応

統計翻訳では、デコーダが構築した複数の翻訳仮説のうち、尤度最大のものを選択することで、適切な翻訳を行う。この尤度は、翻訳に使用するサブモデルが返す値（素性関数値）を、線形または対数線形結合することで算出する。対数線形モデルでは、以下の式で尤度 $\log P(e|f)$ を算出する。

$$\log P(e|f) \propto w \cdot h(e, f)$$

ただし、 $h(e, f)$ は素性ベクトル、 w は素性関数の重みベクトルで、どちらも実数値のベクトルである。たとえば、フレーズベース統計翻訳システムの一つである Moses^[2] のデフォルト設定では、サブモデルとしてフレーズテーブル、言語モデル、語彙化並び替えモデルを用いており、これらを 15 次元の素性ベクトル・重みベ

クトルに配置して、尤度を算出している¹。ドメイン適応は、(a) 重みベクトル w を適応させる方法、(b) 素性関数 $h(e, f)$ を適応させる方法、(c) その他の3種類に大別することができる。

2.1 重みベクトルによる適応

Foster らは、翻訳器が使用するサブモデルを線形補間、対数線形補間で結合する際、重みをドメイン毎に変えることで適応を行った^[3]。彼らはパープレキシティなどを目標関数にして、独自の重み推定を行ったが、近年は、重みの推定に誤り率最小訓練法 (MERT)^[4]などの最適化方法が用いられている^[5]。

素性空間拡張法^[1]は、翻訳に限らず、機械学習全般に使われるドメイン適応方式で、素性関数の重みをドメイン毎に最適化する (3.1 節参照)。Clark らは、これを対数線形補間方式の一種として翻訳に適用し、効果があったと報告している^[6]。なお、彼らは単一のモデルを用い、モデルの重みのみをドメイン適応させている。

2.2 素性関数による適応

素性関数 (素性ベクトル) $h(e, f)$ によるドメイン適応は、さらに、訓練済みサブモデル自身を変更する方法と、コーパスをドメインに適したものに限定しモデルを訓練する方法の2つに分けられる。

2.2.1 モデルのドメイン適応

サブモデル自身を変更する方法には、fill-up 法^[7]、翻訳モデル混合^[8]、インスタンス重み付け^[9]が知られている。これらは、インドメインのフレーズテーブルとアウトドメインのフレーズテーブルを (事前=静的に、あるいはデコード時=動的に) 混合し、新たなフレーズテーブルを生成する。

いずれの方法も、素性関数値のみでなく、デコード時のフレーズ候補も2つのモデルのフレーズを使用するため、一般的には未知語は減少する。しかし、フレーズテーブル以外のサブモデルは、別の構築・混合法を使わなければならないというデメリットも存在する。

1 内訳は、フレーズテーブル 4 次元、歪モデル 1 次元、語彙化並び替えモデル 6 次元、n-gram 言語モデル 1 次元、単語ペナルティ、フレーズペナルティ、未知語ペナルティ各 1 次元である。このうち、未知語ペナルティは重み調整不可とされており、-100 固定である。

2.2.2 コーパスフィルタリング

素性ベクトル $h(e, f)$ を変更するもう一つの方法は、モデル訓練用コーパスをドメイン適応させる方法である。

コーパスフィルタリングのベースは、インドメインとアウトドメインのデータを単純結合して学習し、1つのモデルを構築する方法である (本稿ではコーパス結合方式と呼ぶ)。このコーパス結合モデルを、各ドメインの開発セットで最適化する。コーパス結合方式は、インドメインとアウトドメインの中間的性質を持つモデルが作られるため、一般的には対象ドメインに対するパラメータの精度は低下する。一方、カバーする語彙が増加 (未知語が減少) するため、インドメインのみで学習されたモデル (単独モデルと呼ぶ) より翻訳品質が向上する場合もある。最終的に翻訳品質が向上するか否かは、未知語の減少とパラメータの精度低下のトレードオフになる。

素性関数を、よりドメインに適応させるには、アウトドメインコーパスから訓練文を取捨選択した方がよい。Axelrod らは、インドメインに適した訓練文を、修正 Moore-Lewis フィルタリングと呼ぶ方法でアウトドメインコーパスから取捨選択した^[10]。選択された文をインドメインコーパスに追加してモデルを訓練することで、ドメインに適応したモデルを構築した。

コーパスフィルタリングは、フレーズテーブルのみでなく、翻訳器が使用する全サブモデルを適応させることができる点がメリットであるが、最適な追加文数をあらかじめ予想できない点がデメリットである。

2.3 その他

その他の方法としては、2つの翻訳器を直列に接続し、アウトドメイン用翻訳器による翻訳結果を、さらにインドメイン用翻訳器を使って訂正する方法もある^[11]。これはドメイン依存訳の生成を一種の誤り訂正ととらえていることに相当する。

3 マルチドメイン適応方式

本節では、我々が提案するマルチドメイン適応方式について説明する。

3.1 素性空間拡張法

素性空間拡張法は、機械学習における素性の重みのドメイン適応に用いられる方式である^[1]。素性空間を共通、インドメイン（ターゲットドメイン）、アウトドメイン（ソースドメイン）に分割し、素性を、それが由来するドメインごとに異なる素性空間に配置し、全体を最適化する。インドメインの素性は共通とインドメイン空間に、アウトドメインの素性は共通とアウトドメイン空間にコピーして配置する。こうすることによって、共通空間にある素性が、イン・アウトお互いに欠落した素性を補完するため、精度が向上する。

通常は、アウトドメインをインドメインに適応するために使用されるが、素性空間拡張法では、アウトドメインとインドメインを同等に扱っており、容易にDドメインに拡張することができる。その場合、素性空間は共通、ドメイン1, ... ドメインDのように、D+1空間に分割される（図1）。この素性マトリクスを最適化し、重みベクトルを得る。

4節の実験では、Moses ツールキットのデフォルト素性を使用する。これを素性空間拡張すると、共通空間では15次元、ドメイン依存空間では各14次元（調整可能な素性のみ）となる。

3.2 提案法

3.2.1 コーパス結合モデルと単独ドメインモデルの導入

機械翻訳では、素性の重みより素性関数の方が翻訳品質に対する影響が大きいいため、素性空間によってモデルを切り替えるのは自然な拡張である。本稿では、全ドメインの素性を配置する共通空間に対応するモデルとして、やはり全ドメインのデータを使って構築したコーパス結合モデルを、1つのドメインの素性だけを配置するドメイン依存空間に対応するモデルとして、やはり1ドメインのデータだけを使って構築した単独ドメインモデルを使用する。具体的には、

- フレーズテーブル、語彙化並び替えモデル、言語モデルなど、すべてのサブモデルについて、当該ドメインのコーパスから単独ドメインモデルを、全ドメインのコーパスからコーパス結合モデルを作成する。
- 素性空間拡張では、共通空間にはコーパス結合モデルのスコアを素性関数値として配置し、各ドメインの空間には、単独ドメインモデルのスコアを配置し、最適化する（図1）。
- デコーディングの際は、まず、単独ドメインモデルとコーパス結合モデルのフレーズテーブルをOR検索し、翻訳仮説を生成する。探索の際には、共通空間と対象ドメインの空間の素性だけを使って尤度計算をする。

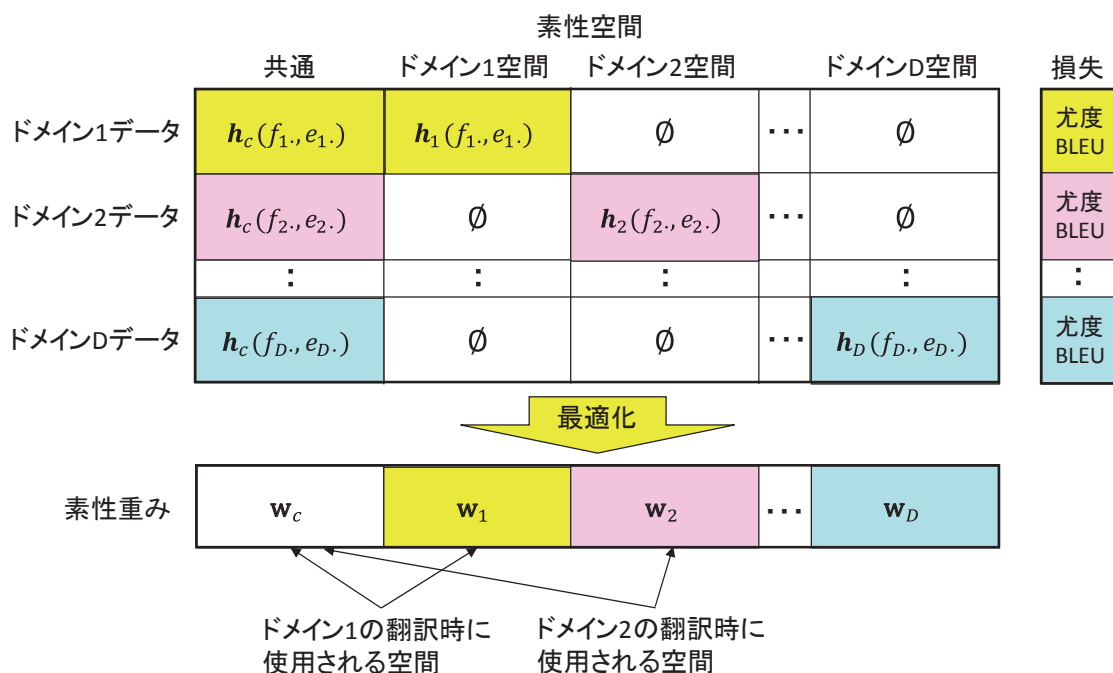


図1 素性空間拡張とコーパス結合モデル/単独ドメインモデルの併用

翻訳仮説生成にコーパス結合フレーズテーブルを使用することにより、他のドメインで出現した翻訳対も利用でき、未知語の減少が期待できる。また、フレーズなどのインスタンスが単独ドメインモデルに存在している場合、高い精度の素性関数値が得られる。このような拡張素性空間を最適化することによって、ドメイン適応を行っているため、翻訳に使用する全サブモデルについて、モデルの種類を問わず適応させることができる。

3.2.2 empty 値

本方式では、コーパス結合モデル、単独ドメインモデルのどちらか一方にのみ出現するフレーズ対が多数存在する。これらフレーズに関しても素性関数は値を返す必要がある。この値を本稿では empty 値と呼ぶ。これはいわば n-gram 言語モデルにおける未知語確率に相当するものであるため、フレーズの翻訳確率分布から算出されるべきものであるが、本稿では、パイパーパラメータとして扱い、開発コーパスにおける BLEU スコアが最高になるよう、実験的に設定する²。

3.3 最適化

3.3.1 同時最適化

一般的な機械学習における素性空間拡張法の利点の一つは、素性空間を操作しているだけなので、最適化アルゴリズムは既存の方法が使えるという点である。

機械翻訳の場合、最適化方法には、誤り率最小訓練 (MERT)^[4]、ペアランク最適化 (PRO)^[12]、k ベストバッチ MIRA (KBMIRA)^[13] が知られている。本稿では、高次元の最適化に適した KBMIRA を使用する。

通常の機械学習における最適化と、機械翻訳の最適化の大きな相違点は、多くの機械学習の損失関数が、尤度などデコーダが出力するスコアを使用しているのに対して、機械翻訳は BLEU^[14] のような、翻訳文の自動評価値を使用する点である。この自動評価値は、翻訳文と参照訳との比較によって算出され、コーパス単位に計算される場合が多い。実際、MERT、KBMIRA は開発セットの BLEU スコアを損失関数の一部に使用している。つまり、複数ドメインを同時に最適化する場合は、ドメ

イン毎に BLEU スコアを算出しないと、結果がドメイン最適にならないことを意味している。

上記問題を解決するため、本稿では KBMIRA を変更する。文献 [13] のアルゴリズム 1 に対する変更点は、以下のとおりである。

- 処理済み翻訳文の BLEU 統計量 (ngram 一致数など) を保存する変数 BG を、1 つからドメイン数 D 個に拡張する。
- 各翻訳文の BLEU スコアは、その翻訳文のドメイン i の BG_i から算出する。
- 素性重みを更新後、その翻訳文の BLEU 統計量を BG_i に追加する。

この変更によって、各ドメイン空間の素性重みは、そのドメインの開発セットに最適化される。

3.3.2 個別最適化

同時最適化は、適応させたいドメインに限られている場合にも、すべてのドメインを最適化しなければならないので、少々オーバースペックである。そこで、全 D ドメインのうち、ドメイン i だけ適応させたい場合、ドメイン i に関連する空間だけに限り、最適化を行う。これを本稿では個別最適化と呼ぶ。

個別最適化は、素性空間を共通空間とドメイン i 空間に制限し、チューニングデータもドメイン i に関するものだけを使用する (図 2)。これは、一般的な対数線形モデルであるので、同時最適化を行わなくても、既存の最適化器をそのまま使うことができる。また、デコーダも、(1) 複数モデルを同時に使えること、(2) empty 値を設定できること、の 2 点を満たすものであればよい。既存のものを少し修正するだけで利用可能となる。同時最適化に比べると、共通空間の最適化が弱くなる恐れがあるが、もともと機械翻訳は素性の重みより素性関

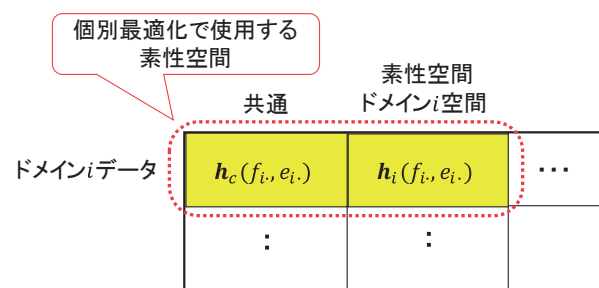


図 2 個別最適化で使用する素性空間

² Moses では、empty 値として一律に -100 を割り当てている。4.2 節で述べるように、これは小さすぎる値で、BLEU スコアが下がる原因となる。

数の影響の方が大きいいため、実用上の問題は少ないと考えられる。ただし、共通空間の素性にコーパス結合モデルを使うこと、empty 値を適切に設定する点は同時最適化と共通である。

4 実験

4.1 実験設定

4.1.1 ドメイン/コーパス

本稿では、英日/日英翻訳を対象に、以下の4つのドメインの最適化を行う。各ドメインのコーパスサイズを表1に示す。MED コーパスは比較的小規模で、それ以外のドメインは100万文規模である。なお、訓練文は80単語以下のものだけを使用した。

- MED: 病院等における医師(スタッフ)と患者の疑似対話のコーパス。内部開発。
- LIVING: 外国人が日本に旅行や在留する際の疑似対話コーパス。内部開発。
- NTCIR: 特許コーパス。訓練コーパスと開発コーパスはNTCIR-8、テストコーパスはNTCIR-9のものを使用³。
- ASPEC: 科学技術文献コーパス^{[15]4}。ASPEC-JEのうち、対訳信頼度の高い100万文を使用。

表1 コーパスサイズ

ドメイン	文数		
	訓練	開発	テスト
MED	222,945	1,000	1,000
LIVING	986,946	1,800	1,800
NTCIR	1,387,713	2,000	2,000
ASPEC	1,000,000	1,790	1,784

4.1.2 翻訳システム

各コーパスの対訳文は、内部開発の事前並べ替えを適用したのちに使用した。翻訳システムの訓練のうち、フレーズテーブル、語彙化並び替えモデルの学習には Moses ツールキット^[2]をデフォルト設定で使用した。言語モデルは KenLM^[16]を用いて、訓練セットの目的言語側から5グラムモデルを構築した。最適化は3.3.1

節で述べたマルチドメインKBMIRAを使用した。デコーディングには、内部開発の Moses のクローンデコーダを使用した。デコーダの設定値は Moses のデフォルト値と同じである。

4.1.3 Empty 値

3.2.2 節で述べた empty 値は、-3 から -20 の間の整数値のうち、全ドメインの開発セット翻訳結果を1ドキュメント扱いしたときの BLEU スコアが最高値になる値を採用した。結果、英日翻訳に関しては -7、日英翻訳は -6 となった。これを確率値とみなした場合、英日は $\exp(-7) \approx 0.0009$ 、日英は $\exp(-6) \approx 0.0025$ となる。

4.1.4 評価指標

評価指標には BLEU を使用し、MultEval ツール^{[17]5}で有意差検定を行った。危険率は $p < 0.05$ とした。最適化の揺れを吸収するため、5回最適化を実施し、その平均値を使用した。

4.1.5 比較方式

各ドメインコーパスだけでモデルを構築、最適化、テストする単独ドメインモデル方式をベースラインにし、他の方式と比較する。従来法としては、2節で述べた以下の方式を使用する。

- コーパス結合: 全ドメインのコーパス結合モデルを使用し、各ドメインの開発セットで最適化、テストした場合。
- 素性空間拡張法 (Clark): 共通空間、ドメイン空間共に、コーパス結合モデルの素性関数を使った素性空間拡張法。文献 [6] の設定と同じだが、最適化にはマルチドメイン KBMIRA を使用した。
- Fill-up 法: ドメイン適応方式に fill-up 法^[7]を用いた場合。
- 翻訳モデル混合: ドメイン適応方式に翻訳モデル混合^[8]を用いた場合。Moses 付属の tmcombine プログラムで混合した。
- コーパスフィルタリング: 文献 [10] で提案された修正 Moore-Lewis フィルタリングで、自分以外の3

3 <http://research.nii.ac.jp/ntcir/index-ja.html>

4 <http://lotus.kuee.kyoto-u.ac.jp/ASPEC/>

5 <https://github.com/jhclark/multeval>

ドメイン（アウトドメイン）のコーパスから対訳文を選択し、対象ドメインの訓練文に加えた場合。追加文数は、100k 文単位に文数を変え、各ドメインの開発セットの BLEU スコアが最高になった文数を採用した。

提案法は、以下のバリエーションをテストする。

- 提案法（同時最適化）：提案法のうち、3.3.1 節で述べた同時最適化を使用した場合。
- 提案法（個別最適化）：提案法のうち、3.3.2 節で述べた個別最適化を使用した場合。
- 提案法（empty=-100）：提案法の empty 値を、Moses と同じ -100 に設定した場合。なお、最適化は個別最適化を使用したか、同時最適化でも同じ傾向が観察された。
- 提案法（アウトドメイン）：提案法のうち、共通空間に対応するモデルとして、コーパス結合モデルではなく、アウトドメインコーパス（インドメインを含まない）だけから学習したモデルを使用した場合。これも個別最適化を使用した。

4.2 翻訳品質

各方式について、英日翻訳および日英翻訳における BLEU スコアを表 2 に示す。なお、表中太字は方式間最高値、赤の値は、単独ドメインモデル方式をベースラインとしたとき有意に向上したもの、青の値は有意に悪化したものを表す ($p < 0.05$)。

単独ドメインモデルと比較した場合、コーパス結合方

式は、翻訳品質が低下する傾向が強かった。素性空間拡張法（Clark）でも同様で、コーパス結合モデルだけを使う方式は、単独ドメインモデルより翻訳品質が下がった。コーパス結合方式は各ドメインが平均化されたモデルが作成され、素性関数の精度が落ちたためと、単独ドメインモデル自体が比較的大きな対訳コーパスから構築されており、単独でも翻訳品質が確保できたためと考えられる。

Fill-up 法は、コーパス結合方式に比べると翻訳品質は向上する場合が多かったが、単独ドメインモデルより悪化した。翻訳モデル混合は、ドメインによって有意に向上する場合と悪化する場合があります、この実験での有効性は確認できなかった。コーパスフィルタリングは、ASPEC の英日翻訳を除き、単独ドメインモデルより有意に向上、または同等品質となった。ASPEC 英日翻訳は 100k 文を加えただけだったが、追加したことが悪影響しており、コーパスフィルタリングは有効だが、最適な追加文数の決定は難しいことを示している。

一方、提案法は、同時最適化、個別最適化ともに、すべてのドメインにおいて単独ドメインモデルより向上あるいは同等品質となり、適切に適應できた。同時最適化に比べ、個別最適化の方が、BLEU スコアが高い傾向がある。MED ドメインは他のドメインに比べてコーパスが小さいため、英日、日英ともに翻訳品質が有意に向上した。他のドメインについては、100 万文規模の大規模コーパスにも関わらず、適應により品質が下がることなく、同等を保った。提案方式は、訓練コーパスサ

表 2 方式別 BLEU スコア

方式	英日翻訳				日英翻訳			
	MED	LIVING	NTCIR	ASPEC	MED	LIVING	NTCIR	ASPRC
単独ドメインモデル	23.23	24.56	38.62	32.69	17.38	19.71	33.63	21.75
コーパス結合	22.65	22.99	38.09	30.59	17.07	18.80	33.21	20.41
素性空間拡張法 (Clark)	22.49	23.97	38.09	30.65	16.75	18.95	33.24	20.39
Fill-up 法	22.42	23.38	38.37	31.50	16.56	19.06	33.14	20.98
翻訳モデル混合	23.81	24.05	38.32	31.97	17.55	19.99	33.32	21.16
コーパスフィルタリング	24.02	24.50	38.77	32.57	18.14	19.76	33.73	21.72
提案法 (同時最適化)	23.69	24.43	38.72	32.69	18.14	20.16	33.68	21.75
提案法 (個別最適化)	23.75	24.43	38.83	32.76	18.43	20.17	33.70	21.81
提案法 (empty=-100)	23.66	23.91	38.56	32.62	17.13	19.86	N/A	N/A
提案法 (アウトドメイン)	23.79	24.29	38.65	32.72	17.32	19.66	33.52	21.73

イズの変化に対して頑健であるといえる。

なお、empty 値を -100 にすると、最適化時に BLEU スコアが振動して、最適化できない場合があった（表中の N/A）。また、提案法でも、共通空間に使用するモデルを、アウトドメインモデルにすると、大部分のケースでは提案法に比べ品質が悪化した。共通空間に使用するモデルは、インドメインを含むコーパス結合モデルの方が望ましい。

まとめると、提案法はドメイン適応方式の中ではほぼ最高品質を確保できた。特に、個別最適化方式のような標準的な対数線形モデルであっても、適切な設定をすれば、ほぼ最良のドメイン適応が実現できることを示している。

4.3 シングルドメイン適応としての効果

ドメイン適応が必要となる場面は、新たなドメインデータの翻訳を行わなければならないにも関わらず、十分な量の訓練文が集まらない場合である。本節では、MED 英日翻訳に絞って、訓練コーパスのサイズを変えて翻訳品質を測定する。

表 3 は、単独ドメインモデル、コーパス結合と提案法（個別最適化）を比較した結果である。表中の赤の値は単独ドメインモデルと比較して有意に高く、青は有意に低いことを表す。緑の値は、単独ドメインモデル、コーパス結合双方と比較して有意に高いことを表している（ $p < 0.05$ ）。

表 3 訓練コーパスサイズ別 BLEU スコア
(MED コーパス、英日翻訳)

訓練文数	単独ドメインモデル	コーパス結合	提案法 (個別最適化)
1k	6.42	17.51	17.59
3k	8.99	17.52	17.95
10k	12.54	18.19	19.02
30k	16.49	19.18	20.28
100k	20.63	20.92	22.53
223k(全部)	23.23	22.65	23.75

訓練コーパスが 1,000 文 (1k) しかない場合は、提案法は単独ドメインモデルに比べて非常に高い品質となっているが、コーパス結合とはほぼ同じである。訓練コーパスサイズが増えるにしたがい、全方式ともに

BLEU スコアが向上するが、コーパス結合の品質向上は単独ドメインモデルより緩やかで、10 万文 (100k) で単独ドメインモデルの品質が逆転する。提案法は、3 千文 (3k) 以上では常に単独ドメインモデル、コーパス結合の品質を上回っており、両者の利点をうまく融合させた方式となっている。

5 まとめ

本稿では、複数ドメインを前提とした、統計翻訳の適応方式を提案した。本稿の方式は、カバレッジが広い(未知語が少ない) コーパス結合モデルと、素性関数の精度がよいドメイン依存モデルを併用し、機械学習分野のドメイン適応方法である、素性空間拡張法の考え方を利用して両者のいいとこどりを行った。また、empty 値をチューニング対象に追加した。

実験では同時最適化を行った場合、個別最適化を行った場合ともに、単独ドメインモデルに比べ、翻訳品質が向上または同等を保持した。提案法は、当該ドメインの訓練コーパスが小規模である場合に高い効果を持ち、100 万文規模の大規模コーパスに適用しても、翻訳品質を下げることなく、ドメインによっては品質向上の効果がある。基本的な対数線形モデルでも、モデルの選択とチューニングを慎重に行うことで、ほぼ最良な適応方式になることを示した。

提案方式は、対数線形モデルに基づく統計翻訳にはすべて適用可能である。今後は、フレーズベースに限らず、木構造変換のような翻訳方式でも効果を確認したい。

謝辞

本研究は総務省の情報通信技術の研究開発「グローバルコミュニケーション計画の推進 - 多言語音声翻訳技術の研究開発及び社会実証 - I. 多言語音声翻訳技術の研究開発」の一環として行われました。

参考文献

- [1] Hal Daumé, III. 2007. Frustratingly easy domain adaptation. In Proc. of ACL, pages 256-263.
- [2] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In Proc. ACL Demo and Poster Sessions, pages 177-180.
- [3] George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In Proc. of WMT. pages 128-135.
- [4] Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In Proc. of ACL, pages 160-167.
- [5] George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In Proc. of EMNLP, pages 451-459.
- [6] Jonathan H. Clark, Alon Lavie, and Chris Dyer. 2012. One system, many domains: Open-domain statistical machine translation via feature augmentation. In Proc. of AMTA.
- [7] Arianna Bisazza, Nick Ruiz, and Marcello Federico. 2011. Fill-up versus interpolation methods for phrase-based SMT adaptation. In Proc. of IWSLT.
- [8] Rico Sennrich. 2012. Perplexity minimization for translation model domain adaptation in statistical machine translation. In Proc. of EACL, pages 539-549.
- [9] George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In Proc. of EMNLP, pages 451-459.
- [10] Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In Proc. of EMNLP, pages 355-362.
- [11] Serena Jeeblee, Weston Feely, Houda Bouamor, Alon Lavie, Nizar Habash, and Kemal Oflazer. 2014. Domain and dialect adaptation for machine translation into egyptian arabic. In Proc. of EMNLP, ANLP Workshop, pages 196-206.
- [12] Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In Proc. of EMNLP, pages 1352-1362.
- [13] Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In Proc. of NAACL-HLT, pages 427-436.
- [14] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In Proc. of ACL, pages 311-318.
- [15] Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. ASEPC: Asian scientific paper excerpt corpus. In Proc. of LREC.
- [16] Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In Proc. of ACL: Short Papers, pages 690-696.
- [17] Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In Proc. of ACL-HLT, pages 176-181.