Japio

依存関係確率モデルを用いた統計的句アライメント

京都大学大学院情報学研究科教授
黒橋　禎夫

PROFILE
1994 年京都大学大学院工学研究科電気工学第二専攻博士課程修了。博士（工学）。2006 年 4 月より京都大学大学院情報学研究科教授。自然言語処理、知識情報処理の研究に従事。

kuro@i.kyoto-u.ac.jp　TEL 075-753-5344

京都大学大学院情報学研究科
中澤　敏明

PROFILE
2006 年東京大学大学院情報理工学系研究科電子情報学専攻修士課程修了。現在京都大学大学院情報学研究科博士後期課程在学。機械翻訳の研究に従事。

1 はじめに

日本語と英語のように言語構造が著しく異なり、語順変化が大きな言語対において、対訳文をアライメントする際に重要なことは二つある。一つは構文解析や依存構造解析などの言語情報をアライメントに組み込み、語順変化を克服することであり、もう一つはアライメントの手法が 1 対 1 の単語対応だけでなく、1 対多や多対多などの句対応を生成できることである。これは一方の言語では 1 語で表現されているものが、他方では 2 語以上で表現されることが少なくないからである。しかしながら、既存のアライメント手法の多くは文を単純に単語列としてしか扱っておらず [1]、句対応は単語対応を行った後にヒューリスティックなルールにより生成するといった方法を取っている [3]。Quirk ら [6] はアライメントに構造情報を統合しようとしたが、前述の単語列アライメントを行った後に用いるに留まっている。単語列アライメント手法そのものの精度が高くないため、このような方法では十分な精度でアライメントが行えるとは言い難い。

一方で、アライメントの最初から構造情報を利用する手法もいくつか提案されており、我々が提案する手法もその一つである。提案手法のポイントは以下の 3 つである。

1. 両言語とも依存構造解析し、アライメントの最初から言語の構造情報を利用する
2. アライメントの最小単位は単語だが、モデル学習時に句となるべき部分を自動的に推定し、句アライメントを行う
3. 各方向（原言語→目的言語と目的言語→原言語）の生成モデルを二つ同時に利用することにより、より高精度なアライメントを行う

既存の構造情報を利用する手法において、これらの 3 つを全て満たすものは提案されておらず、我々の手法はこの点で新規性がある。

本モデルは二つの依存構造木において、一方の依存構造木で直接の親子関係にある一組の対応について、他方のそれぞれの対応先の依存関係をモデル化しており、単語列アライメントで扱うのが困難な距離の大きな語順変化にも対応することができる。言い替えれば、本モデルは木構造上での reordering モデルということができる。また本モデルはヒューリスティックなルールを用いずに、句となるべき部分を自動的に推定することができる。ここでいう句とは必ずしも言語的な句である必要はなく、任意の単語のまとまりである。ただし、Phrase-based SMT における句の定義との重要な違いは、我々は木構造を扱っており、単語列としては連続でなくても、木構造上で連続ならば句として扱っているという点である。

また我々のモデルは IBM モデルのような各方向の生成モデルを両方向分同時に用いてアライメントを行う。これはアライメントの良さを両方向から判断する方が自然であり、Liang ら [4] による報告にもあるように、そうした方が精度よいアライメントが行えるからである。ただし、Liang らの手法が IBM モデルと同様に単語列を扱うものであるのに対し、提案手法は木構造を扱っているという重要な違いがある。また Liang らの手法では部分的に双方向のモデルを結合するに留まっており、アライメントの結果としては各方向それぞれ独立に生成されるが、我々の方法ではただ一つのアライメントを生成するという違いもある。

最近の報告では生成モデルよりも識別モデルを用いた方がより高精度なアライメントが行えるという報告がなされているが、学習用にアライメントの正解セットを用意するコストがかかってしまう。そこで我々は教師なしでモデル学習が行える生成モデルを用いた。モデルは 2 つのステップを経て学習される。Step 1 では単語翻訳確率を学習し、Step 2 では句翻訳確率と依存関係確率が推定される。さらに Step 2 では単語対応が句対応に拡張される。各 Step は EM アルゴリズムにより反復的に実行される。

2 提案モデル概観

以降の説明においては言語対として日本語と英語を用いるが、提案モデルはこの言語対に特別に設計されたものではなく、言語対によらないロバストなものである。

提案モデルは依存構造木上で定義されるものであるので、まず対訳文を両言語とも依存構造解析し、単語の依存構造木に変換する。図 1 に依存構造木の例を示す。単語は上から下に順に並んでおり、文のヘッドとなる単語は最も左側に位置している。アライメントの最小単位はこれら各単語であるが、モデル推定時に複数単語のかたまりを句として自動的に獲得する。これについては 3.2 章で詳しく述べる。

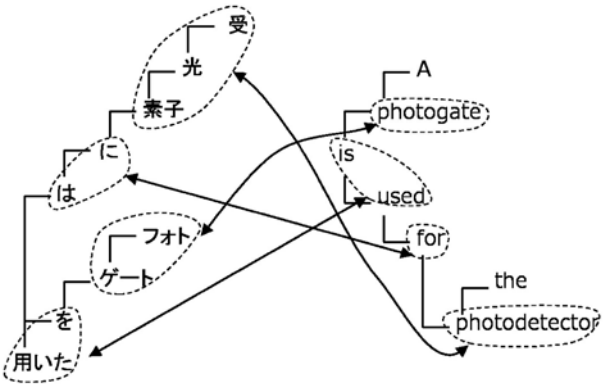

図 1　依存構造木とそのアライメント例

IBM モデル [1] では、与えられた日本語文 f と英語文 e からなる対訳文間の最も良いアライメント $\hat{\mathbf{a}}$ は以下の式により獲得される：

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} p(\mathbf{f}\,|\,\mathbf{a},\mathbf{e}) \cdot p(\mathbf{a}\,|\,\mathbf{e}) \qquad (1)$$

ここで $p(\mathrm{f}|\mathrm{a},\mathrm{e})$ は語彙確率 (lexicon probability) と呼ばれ、各単語の翻訳確率の積である。$p(\mathrm{a}|\mathrm{e})$ はアライメント確率 (alignment probability) と呼ばれ、前後の単語との位置関係をモデル化するものである。IBM モデルは方向性があるため、アライメントに制限がある。これを解消するため、両方向による結果を最後に統合して最終的なアライメントとすることが多い。しかし日英のような言語構造の違いの大きい言語対においては、このような方法では十分な精度でのアライメントは行えない。

提案モデルは IBM モデルを 3 つの点で改善する。一つ目は単語ではなく句を考慮すること、二つ目は文中での単語の位置ではなく、依存関係を考慮すること、最後に最も良いアライメント $\hat{\mathbf{a}}$ を求める際に、片方向のモデルだけでなく、両方向のモデルを同時に利用する。つまり、式 (1) を以下のように変更する：

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} p(\mathbf{f}\,|\,\mathbf{a},\mathbf{e}) \cdot p(\mathbf{a}\,|\,\mathbf{e}) \cdot p(\mathbf{e}\,|\,\mathbf{a},\mathbf{f}) \cdot p(\mathbf{a}\,|\,\mathbf{f})$$

提案モデルは EM アルゴリズムにより学習される。なお各確率の定義は [8] を参照されたい。

3 モデルトレーニング

提案モデルは２つのステップに分けて学習される。これは IBM モデルにおいて、完全に最適解が求まる簡単なモデルからスタートし、徐々により複雑なモデルに移行することに対応する。Step 1 では単語翻訳確率の推定が行われ、Step 2 では句翻訳確率と依存関係確率の推定が行われる。どちらのステップにおいてもモデルは EM アルゴリズムにより学習される。またステップ 1 において句は扱わず、全て単語単位での学習となる。複数単語の塊＝句は Step 2 において自動的に獲得される。

3.1 Step1

Step 1 では各方向独立に、単語翻訳確率を推定する。これは IBM Model 1 と全く同様の方法により行われる。Step 1 の推定の際には対応の単位は各ノード単体、つまり単語のみであり、句は考慮しない。句は Step 2 の推定から考慮し、句となるべき候補を動的に作り出すことにより実現する。これは Step 1 の段階で可能な句の候補全てを考慮すると、アライメント候補数が爆発し、扱えなくなるためである。

3.2 Step2

Step 2 では句翻訳確率と依存関係確率の両方を推定する。また f から e、e から f の二つのモデルを同時に用いて、一つの方向性のないアライメントを得る。Step 1 では計算を効率化することにより、近似を用いずにモデルの推定が完全に行えるが、Step 2 では可能なアライメントを全て考慮することは不可能である。そこで我々は最も良いアライメントを探索するために、まず句翻訳確率のみから初期アライメントを生成し、その後依存関係確率も考慮しつつ、山登り法によってアライメントを徐々に修正するという方法をとる。

さらに Step 2 において新たな句候補の生成を行う。新たな句候補は山登り法によって求められた最も良いアライメントの状態から生成され、次のイタレーションから考慮される。つまり、Step 2 のイタレーションが進むに連れ、より大きな句の対応を発見することができる。

全体として、Step 2 の 1 回のイタレーションは、E-step での `` 初期アライメント '' の生成と `` 山登り法 '' により最適なアライメントの探索、E-step と M-step の間での新たな句候補の生成、M-step でのパラメータの更新の４つの要素からなる。

Step 2 での一回目のイタレーションでは、パラメータの初期値を以下のようにする。一回目のイタレーションにおいては全ての句は 1 単語からなるため (2 単語以上からなる句候補が獲得されていないため) 句翻訳確率については、Step 1 で求めた単語翻訳確率をそのまま用いる。依存関係確率は、Step 1 の最後のイタレーションで得られた最も良いアライメント結果において依存関係の生起回数を計数し、そこから求めた確率を用いる。

初期アライメント (E-step)：

依存関係確率は用いず、句翻訳確率のみから初期アライメントを生成する。全ての句候補同士の対応 (もしくは NULL 対応) に対して、各方向からの句翻訳確率を掛け合わせた、句対応確率を計算し、句対応確率の高いものから順に、対応として採用する。この際、各単語は 1 度しか対応付かないようにする。つまりすでに採用されている対応と重なるような対応は採用しない。なお句候補の生成については後で述べる。

初期アライメントが生成されたら、その状態でのアライメント確率を計算する。このときから依存関係確率も用いる。

山登り法 (E-step)：

初期アライメントの状態から、依存関係確率を考慮しながらアライメントを修正し、徐々に確率の高いアライメントを探索していく。修正手段としては以下の４種類を考える。

・Swap：任意の２つの対応に注目し、それらの対応を入れ替える。例えば図２の最初の操作では、`` 光 ⇔ photogate" と `` フォト ゲート ⇔ photodetector" の対応がそれぞれ `` 光 ⇔ photodetector" と `` フォト ゲート ⇔ photogate" というように対応が入れ替

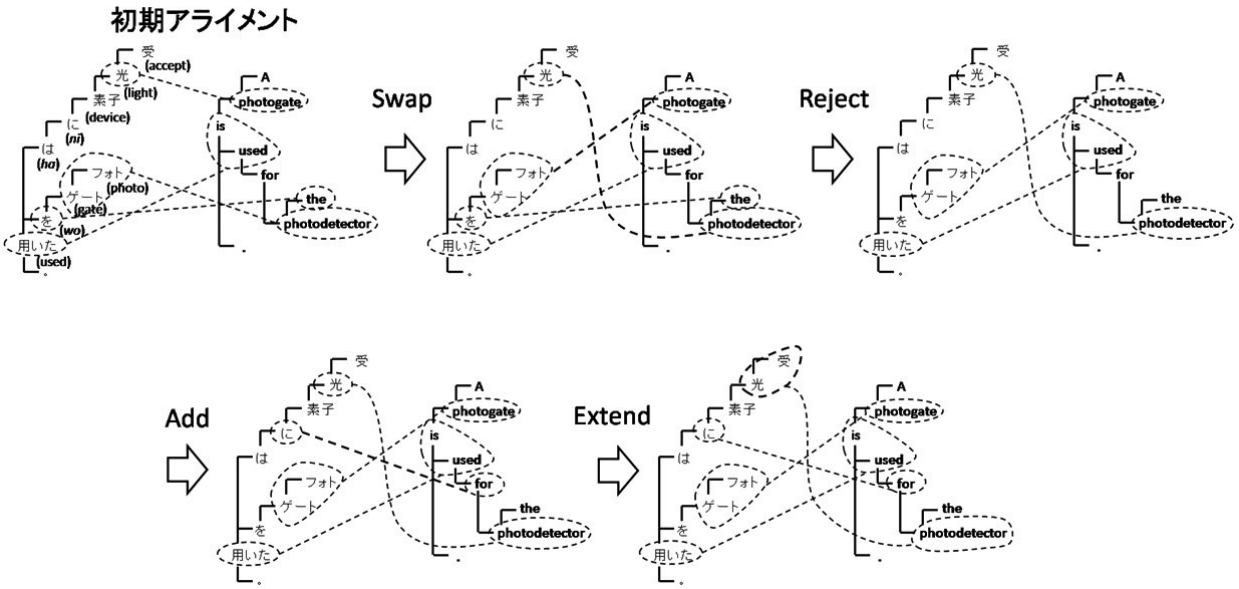

図２　山登り法の例

えられている。

・Extend：任意の 1 つの対応に注目し、そのいずれかの言語における句を、親または子方向に 1 ノード分だけ拡大する。

・Add： NULL 対応となっている原言語側及び目的言語側のノード間に、新たに対応を追加する。

・Reject：すでにある対応を削除し、それぞれ NULL 対応とする。

図２に山登り法によるアライメント修正過程の例を示す。なお図２は 1 回以上イタレーションを行ったあとの状態である。修正後のアライメント確率が修正前よりも高くなる場合にのみ修正を実行し、修正された状態から再度修正を行っていく。確率が高くなる修正箇所がなくなるまで修正を繰り返し行い、最終的に得られたアライメントが、最も確率の高いアライメントとなる。なお修正の途中で得られたアライメントの状態を、確率の高いものから n 個保存しておき、仮想的な n-best アライメントとし、パラメータ推定の際に利用する。

新たな句候補の生成：

山登り法により得られた最も良いアライメント結果のうち、NULL 対応となった単語に注目する。NULL 対応となった語の親、または子の単語が NULL 対応でなければ、その単語と NULL 対応の単語とをまとめたものを新たに句として獲得し、Step ２の次のイタレーションから探索範囲に入れる。例えば図２の最終状態において NULL 対応となっている `` 素子 '' は、その子の対応である `` 受 光 ⇔ photodetector'' に含まれ、新たに `` 受 光 素子 '' という句を作りだし、`` 受 光 素子 ⇔ photodetector'' という対応があるものと考えるさらに親の対応である `` に ⇔ for'' に含まれ、`` 素子 に '' という句もつくり出し、`` 素子 に ⇔ for'' という対応があるものと考える。これらの新たに考慮される対応には、元の対応の出現期待値 (正規化されたアライメントの確率) を分配する。

このように、NULL 対応に注目することにより動的に句となるべきかたまりを獲得していき、モデルの構築を行う。

モデル推定 (M-step)：

一般的な EM アルゴリズムにおいては、得られた n-best アライメントのそれぞれのアライメント確率を正規化し、各アライメントにおけるパラメータの出現回数をこの正規化された確率値 (出現期待値) を用いて計数する。我々もこの方法に従い、全ての対訳文での全てのアライメント結果を集めてパラメータの推定を行う。ただし、正確に全てのアライメントを数え上げることはできないため、山登り法の途中で得られたアライメント

のうち、アライメントの確率の高いもの上位 n 個 (山登りの回数が n に満たない場合はその全て) を用いる。パラメータ推定は一般的な EM アルゴリズムと同様に、各パラメータの出現期待値の総和を全体の回数で正規化することにより行われる。

ここまでの処理により、EM アルゴリズムの E-step、M-step が終了し、再び E-step に戻る。これを複数回繰り返すことにより、モデルのトレーニングを行う。

4 実験

提案手法の有効性を示すためにアライメント実験を行った。トレーニングコーパスとして JST 日英抄録コーパスを用いた。このコーパスは、科学技術振興機構所有の約 200 万件の日英抄録から、内山・井佐原の方法 [7] により、情報通信研究機構が作成したものであり、100 万対訳文からなる。このうち 475 文に人手で正解のアライメントを付与し、正解データとした。正解データは Sure(*S*) と Possible(*P*) の 2 段階に分けてアライメントされている [5]。また評価の単位は日本語、英語とも単語とし、適合率・再現率・AER により精度を求めた。

日本語文に対しては形態素解析器 JUMAN および依存構造解析器 KNP を用い、英語文に対しては Charniak の nlparser を用いて構文解析し、ルールにより単語の依存構造木に変換する。また Step2 のパラメータ推定の際に用いるアライメントの数は n=10 とした。

比較実験として、単語列アライメント手法として広く利用されている IBM モデルを実装したアライメントツールである GIZA++[5] を用いてアライメントを行った。各モデルのイタレーション回数などのオプションはデフォルトの設定をそのまま利用した。さらに各方向のアライメント結果を三つの対称化手法により統合した。結果を表 1 の下部 3 行に示す。利用した対称化手法は 'intersection'、'grow-final-and'、'grow-diag-final-and' の 3 つである [2]。

	Pre	Rec	AER
Step 1	77.5	33.92	47.20
Step 2-1	84.26	48.38	61.65
Step 2-2	84.85	57.26	68.53
Step 2-3	82.84	61.86	71.03
Step 2-4	80.21	63.13	70.88
Step 2-5	78.71	64.10	70.88
Step 2-6	78.02	63.38	70.76
Step 2-7	76.83	64.60	70.39
Step 2-8	71.99	67.71	69.85
intersection	90.51	45.16	60.31
grow-final-and	79.92	60.06	68.70
grow-diag-final-and	77.80	61.47	68.77

表 1　アライメント実験結果

一方、提案手法によるアライメント精度を表 1 の上部に示す。まず 'Step 1' に示されているのは、Step 1 のイタレーションを 5 回行った後に学習されたパラメータ (単語翻訳確率) を用いたアライメントの精度である。なおここでのアライメントは、両方向のパラメータを用いて、3．2 章の初期アライメント生成手法と同様にアライメントを生成した結果である。'Step 2-X' は Step 2 の各イタレーション終了時点でのアライメント精度である。'Step 2-1' は句翻訳確率は 'Step 1' のものと同じだが、それに加えて 'Step 1' のアライメント結果から推定した依存関係確率を用いてアライメントを行っている。つまり、'Step 1' と 'Step 2-1' とを比較することにより、依存関係確率を用いることによるアライメント精度の向上が見て取れる。以後 Step 2 のイタレーションを行い、その都度アライメント精度を計測した。結果として、提案手法では単語列アライメント手法よりも AER で 2.3 ポイントのアライメント精度向上を達成した (Step 2-3 と grow-diag-final-and との比較による)。適合率だけを見ると 'intersection' が最もよい値を示しているが再現率が極端に低くなっている。また再現率が最も高いのは 'grow-diag-final-and' であるが、同程度の再現率を示している提案手法の結果を見ると、適合率では大きく上回っており、総合的に見

て提案手法は単語列アライメント手法よりも優れているということができる。

5 まとめと今後の課題

本稿では依存関係確率モデルを用いた統計的句アライメント手法を提案した。提案モデルは木構造上でのreordering モデルということができ、シンプルなモデルながらも言語構造の違いを柔軟に吸収し、精度の高いアライメントを実現できた。実験結果から、語順の大きく異なる言語対に対しては既存の単語列アライメント手法では十分な精度を達成することは困難であり、構文解析などの言語情報を利用することが自然であり、高い効果を示すことが証明された。今回は日本語と英語間のアライメント実験のみしか行わなかったが、同様に語順に大きな違いのある日本語と中国語間での実験などを行い、提案手法が言語対によらずロバストな手法であることを示す必要がある。

考察にも述べたとおり、提案手法は依存構造解析に大きく依存しており、依存構造解析誤りが容易にアライメントの誤りにつながってしまう。両言語の解析結果を照らしあわせて、文構造を修正しつつアライメントすることも可能なはずであり、現在検討中である。これが実現できれば、依存構造解析とアライメント双方の精度向上が可能となると考える。

アライメントの精度のみを評価したが、この結果が翻訳の精度にどのように影響するかを調査することは今後の課題である。

参考文献

[1] Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., and Mercer, R. L. (1993). “The Mathematics of Statistical Machine Translation: Parameter Estimation.” Association for Computational Linguistics, 19 (2), pp. 263-312.

[2] Koehn, P., Axelrod, A., Mayne, A. B., Callison-Burch, C., Osborne, M., and Talbot, D. (2005). “Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation.” In Proceedings of International Workshop on Spoken Language Translation 2005 (IWSLT'05).

[3] Koehn, P., Och, F. J., and Marcu, D. (2003). “Statistical Phrase-Based Translation.” In HLT-NAACL 2003: Main Proceedings, pp. 127-133.

[4] Liang, P., Taskar, B., and Klein, D. (2006). “Alignment by Agreement.” In Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, pp. 104-111 New York City, USA. Association for Computational Linguistics.

[5] Och, F. J. and Ney, H. (2003). “A Systematic Comparison of Various Statistical Alignment Models.” Association for Computational Linguistics, 29 (1), pp. 19-51.

[6] Quirk, C., Menezes, A., and Cherry, C. (2005). “Dependency Treelet Translation: Syntactically Informed Phrasal SMT.” In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), pp. 271-279.

[7] Utiyama, M. and Isahara, H. (2007). “A Japanese-English Patent Parallel Corpus.” In MT summit XI, pp. 475-482.

[8] Nakazawa, T. and Kurohashi S. (2009) "Statistical Phrase Alignment Model Using Dependency Relation Probability." In Proceedings of the third Workshop on Syntax and Structure in Statistical Translation (SSST-3), pp.10-18, Colorado, U.S.A (2009.6).