機械翻訳における日本語格助詞の生成

鈴木久美　　Kristina Toutanova

Microsoft Research
One Microsoft Way, Redmond WA 98052 USA
{hisamis, kristout}@microsoft.com

概要

統計的英日機械翻訳システムにおける格助詞の生成を改善するモデルを提案する。このモデルは、従来の統計的機械翻訳システムの各モデルと同様、文対応つきの対訳文書から学習されるが、格助詞の生成に特化したモデルであるため、このタスクに有益である構文的素性・長距離素性が多く利用されている。格助詞生成モデルを最大エントロピー法を用いて構築し、treeletに基づく統計的機械翻訳システム(Quirk et al., 2005)で使用したところ、BLEUスコアによる自動評価、人手による評価ともに、格助詞生成において改善を確かめることができた。

1　はじめに

格助詞をはじめとする、いわゆる機能語（付属語）の生成は機械翻訳においてこれまであまり脚光を浴びてこなかった分野である。たとえば現在主流となっている統計的機械翻訳(statistical machine translation, SMT)は、まず語単位で原言語と目的言語の対応を取っていくところから始まるが、このとき機能語と自立語はまったく同じものとして扱われる。しかし、このような方法は原言語と目的言語が言語的に大きく異なる場合、機能語に関しては非常に不自然である。本来、機能語は自立語の機能を当該言語内部で表示するためのものであり、原言語から直接学びうるものではない。本稿はこうした点から、目的言語である日本語の格助詞の付与を、この目的に特化したモデルを使用することによって改善しようという試みである。

図1は、ベースラインとして使用した、treeletに基づくSMTシステム[6]の出力の例である。自立語に関しては、"the patch"をこのドメインに適した「修正プログラム」と訳出し、また英語の無生物主語他動詞文を日本語らしく受身形を使って翻訳するなど、SMTの強いところがよく出ているが、格助詞に間違いがあるため、「何が」「何で」置き換えられるのかが明確にわからない訳文になってしまっている。このような「惜しい」翻訳の例は枚挙にいとまがなく、格助詞を修正することによって使えるようになる翻訳結果はかなりの数に上ると考えられる。

本稿では、英日翻訳における格助詞の生成を、このタスクに特化したモデルを使用することによって改善する。使用したモデルは、[1,7]で提案した格助詞予測モデルに基づくものであるが、[1]では日本語文の素性しか使用できなかったのに対し、機械翻訳時での格助詞生成では原言語の英文も参照できることから、これも素性の抽出時に使用した。また、treelet SMTでは、原言語・目的言語ともに依存構造を使用しているので、この構造も素性の抽出に用いた。文対応つきのコンピュータ・マニュアル文、490万対を使用して格助詞予測モデルを構築し、おなじドメインのテストデータ2,000文で評価したところ、BLEU[5]で統計的に有意な改善を確かめることができた。また、人手による評価で、この方法は正解を誤りに変換してしまうという副作用が非常に少ない点で、SMTシステムでの実装に適している方法であることもわかった。

S: The patch replaces the .dll file.
O: 修正プログラムを.dllファイルが置き換えられます。
C: 修正プログラムで.dllファイルが置き換えられます。

図1: SMTによる翻訳の例 (S: 翻訳前の文; O: 翻訳結果; C: 正しい翻訳)

2　格助詞生成タスクの定義とモデル

2.1　タスクの定義

タスクの定義は基本的に[1,7]に依っている。すなわち、原言語（英語）と目的言語（日本語）の文対応付きのコーパスから、日本語の格助詞を予測するモデルを構築する。このモデルを、機械翻訳の出力文tに後処理としてかけ、格助詞を生成あるいは削除することによって、tを日本語としてまた元の英文sの翻訳としてより適した形にすることが目的である。評価はBLEUスコアと、人手で行う。またその際、tは、語の選択や語順の点で適切な日本語であるとは到底言えないようなものが多く含まれている可能性が大きいので、語の選択・語順が理想的な状態でのモデルの精度を得るために、モデルを人手による正解翻訳r (reference translation)にかけたときの精度も評価する。

翻訳文t（あるいは人手による翻訳r）は、まず品詞タガーにより品詞を付与し、これに基づいて文節に区切る。格助詞を予測する位置は各文節の終わり（句読点を除く）と定義した。図1の例だと、以下の□に相当する箇所

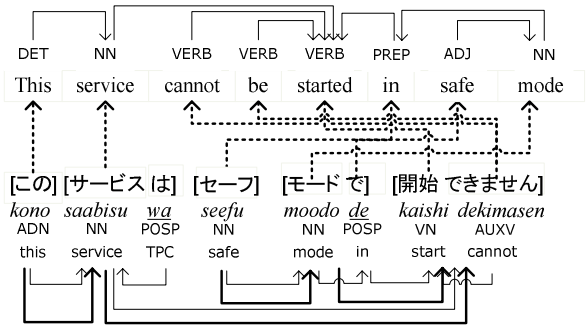

図2: 文・語対応付きの構造の例

に、どの格助詞が適当(あるいはどの格助詞も不適当)かをモデルによって判断する。

[修正□][プログラム□][.dll□][ファイル□][置き換えられます□。]

生成の対象とした助詞は以下の18の格助詞である[1]: は、が、を、の、に、から、と、で、へ、まで、より、には、からは、とは、では、へは、までは、よりは。格助詞が不適当な場合はNONEを出力ラベルとして使用した。したがって、このタスクは各文節を19のクラスに分類するタスクとなる。

2.2 Treeletに基づく翻訳システム

本稿における格助詞生成タスクのSMTへの実装は、treelet翻訳システム[6]を用いて行った。このシステムは、依存構造のサブグラフ(treelet)をもちいて翻訳を進めていく。Treeletシステムで使用される文・語対応付きのデータを図2に示す。文対応付きのデータはまず、GIZA++[3]で語対応をつける(図2の破線部分)。次に、原言語の入力文にパーサを使って依存構造を与える(細い実線部分)。この依存構造のサブグラフを、目的言語側の語と対応させていくのだが、その際、語対応のリンクをたどって、目的言語側にも語レベルでの依存構造が与えられる。ただし、本タスクは文節への格助詞付与を対象にしているため、語レベルの依存構造を文節レベルの依存構造に変換して使用した(太い実線部分)。

翻訳候補は目的言語側のtreeletを組み合わせて生成される。候補のスコアは、素性関数を線形に組み合わせたモデルを使ってランクされる。この線形モデルは

$$score(t) = \sum_j \lambda_j f_j(t)$$

で示される。ただしλ_jはモデルのパラメータ、$f_j(t)$は素性関数jの候補tにおける値である。実験に使用したtreelet SMTは、翻訳モデル、語順モデル、言語モデルなど、10個の素性関数を使用している。本稿では格助詞生成モデルを、11個目の素性関数として使用した。モデルのパラメータ学習は、max-BLEU法[2]を用いて行った。

[1] なかには厳密な意味では格助詞とは言えないものも含む。詳細は[1]を参照。

2.3 格助詞生成モデル

格助詞生成のモデルは、[1,7]で提案した格助詞予測モデルに基づいている。すなわち、ある文の各文節を19のクラス(18の格助詞プラスNONE)に分類する分類器を最大エントロピーモデルを用いて実装した。入力文をs、翻訳文をt、図2に示したような語対応や依存構造などの追加情報をAとし、tの格助詞を除いた部分を$rest(t)$、tに与えられた特定の格助詞列を$case(t)$とすると、モデルの推定する確率は

$$P_{case}(case(t) \mid rest(t), s, A)$$

で表される。ここでは、各文節への格助詞付与を独立の事象とみなしている(Cf. [7])。

実験に使用した素性と、「サービス(は)」という文節に注目したときの素性の値の例を表1に挙げる。これらは素性選択の結果選択された素性をもとにしている(詳細については[8]を参照)。素性は目的言語の語・品詞・依存構造に関するものに加え、語対応のリンクを通じて、原言語の語・品詞・依存構造情報も参照できる。たとえば、表1の"Aligned to parent word POS"は、注目している文節の係り先の文節が対応している語の品詞、という情報である。このように、原言語を参照する素性、品詞情報や依存構造が使われている素性が実際に多く選択されており、利用できる素性は通常のSMTシステムが参照する文脈情報よりも広範・多岐にわたっている。

Features	Example
Words in position –1 and +2	この、モード
Headword & previous headword	サービス&この
Parent word	開始
Aligned word	*service*
Parent of word aligned to headword	*started*
Next word POS	*NOUN*
Next word & next word POS	セーフ*&NN*
Headword POS	*NOUN*
Parent headword POS	*VN*
Aligned to parent word POS & next word POS & prev word POS	*VERB&NN&ADN*
Parent POS of word aligned to headword	*VERB*
Aligned word POS & headword POS & prev word POS	*NN&NN&ADN*
POS of word aligned to headword	*NOUN*

表1: 使用した素性

2.4 人手による翻訳を使ったモデルの評価

上述のモデルの精度の上限を知るため、モデルを理想的な条件下で評価した。理想的な条件とは、翻訳候補に語選択・語順の誤りが含まれていないことであり、これは人手による翻訳文(reference translation)を使って代用できる。モデルのトレーニングには、treeletの構築に使用したものと同じ500万文対から、評価用に1万文を残した残り490万文対を使用し、評価には残しておいた1万文からとった5,000文を使用した。このデータセットの平均文

モデル	ACC	BLEU
ベースライン (frequency)	58.9	40.0
ベースライン (490K LM)	87.2	83.6
提案モデル	94.9	93.0

表2: モデルの正解率(ACC, %)とBLEU

長は英語で約15語、日本語で約19語であった。

表2には、頻度によるベースライン(常に最も高頻度のラベル(=NONE)を選択)と、word trigram言語モデルを用いたベースラインの結果も示した。この結果から、提案手法は与えられた文脈情報をうまく利用し、ノイズの少ない状況下においては、ほぼ95%の精度で格助詞を生成できる、ということがわかった。また、精度の向上がBLEUに反映されることも見て取れる。

3 機械翻訳における格助詞生成

本稿の目的は、上述のモデルを使用して、実際の翻訳の質を高めることである。ここでは、実験の手軽さを考慮して、2.2節で述べたように格助詞モデルを素性関数として使用し、Nベスト解の並べかえ[4]に使用した。素性関数の値には、格助詞生成モデルによる確率の対数を用い、関数の重みはこの目的用の文対応付きコーパス(1,000文)を使って、max-BLEU法で推定した。

Nベスト解の並べかえを格助詞生成モデルに適用するにあたって注意すべきことは、当該Nベスト解の中に格助詞のみが異なる翻訳候補が含まれていない可能性があることである。そのような場合、モデルが翻訳結果を改善する可能性もなくなってしまう。実際に、treeletシステムによるスコアが一番高い解から格助詞のみが異なる候補をモデルを用いて生成し、そのトップ40のうちいくつがNベスト解(n=1,000)の中に含まれているかを、ディベロップメントデータを使って調べてみたところ、その確率は0.023であった。つまり、格助詞のみ異なる解ベスト40のうち、1,000ベスト解に含まれているのは平均ひとつ以下($40 \times 0.023 = 0.92$)しかないことになる。

この点を考慮して、実験ではNベスト解を「拡張」してから使用することにした。すなわち、Nベスト解のそれぞれに対して、格助詞生成モデルで格助詞のみが異なる候補を生成し、そのうちトップk個をNベスト解のリストに加えた[2]。実験で、kは$0 \le k \le 40$に設定した。k =0のとき、通常のNベスト解並びかえと同じタスクになる。

拡張されたNベスト解を使うとき問題になるのは、これらは新規に生成された候補なので、翻訳モデルをはじめとする線形モデルで使われているモデルのスコアを再計算しなくてはならない、ということである。ここでは、言語モデルスコアなど、再計算が容易な素性関数4つのみを再計算し、残りの関数は拡張前の候補のスコアをそのまま使用した。さらに、拡張された候補の特徴をとらえるため、次の4つの素性を新たに追加した[3]。

- **Generated:** 新規生成された候補=1, それ以外=0.
- **NONE→non-NONE:** 格助詞なしからありに変わった文節の総数.
- **Non-NONE→NONE:** 格助詞ありからなしに変わった文節の総数.
- **Non-NONE→non-NONE:** ある格助詞から別の格助詞に変わった文節の総数.

4 実験結果と考察

4.1 評価データ

評価には、SMT・格助詞モデル構築に使用したものと同じドメインの別のデータを用意した。ディベロップメントデータ1,000文はモデルの選択に、テストデータ2,000文は人手評価用に使用した。

4.2 実験結果

BLEUによる実験結果を表3に示す。nはtreeletシステムから得たNベスト解の数、kは格助詞が異なる拡張候補の数であり、Oracle BLEUは、与えられたリストからBLEUの値を最大にするような候補を選んだときのスコアである。一番上の行は格助詞生成モデルを使用しない(n=1, k=0)ときのスコア、すなわちベースラインの結果である。このときのBLEUは約38であった。

n	k	BLEU	Oracle BLEU
1	0	37.99	37.99
20	0	37.83	41.79
100	0	38.02	42.79
1000	0	38.08	43.14
1	1	38.18	38.75
1	10	38.42	40.51
1	20	38.54	41.15
1	40	38.41	41.74
20	10	**38.91**	45.32
20	20	38.72	45.94
20	40	38.78	46.56
100	10	38.73	46.87
100	20	38.64	47.47
100	40	38.74	47.96

表3: 提案モデルの評価結果

次の3行(k=0)は、格助詞の異なる候補の拡張をせずに、通常のNベスト並びかえをしたときの結果である。表からわかるように、n=1,000まで大きくすると、Oracle BLEUは大きく改善するが、BLEUスコアはほとんど改善していない。これに対し、拡張したNベストリストを使用する本稿提案の方法(k>0、次の4列)では、BLEUの改善が見られる。特に、k=1のとき、つまりSMTのスコア最大の解に最適な

[2] このとき、拡張されたリストの候補数は$n(k+1)$である。

[3] これらの素性の値はk=0のときはすべて0である。

		Fluency			Adequacy		
		Annotator #1			Annotator #1		
		S	B	E	S	B	E
Anno-tator #2	S	**27**	1	8	**17**	0	9
	B	1	**9**	16	0	**9**	12
	E	7	4	**27**	9	8	**36**

表3: モデル(*n*=20, *k*=10)の人手による評価

		Fluency			Adequacy		
		Annotator #1			Annotator #1		
		S	B	E	S	B	E
Anno-tator #2	S	**42**	0	9	**30**	1	9
	B	1	**0**	7	0	**9**	7
	E	7	2	**32**	9	3	**32**

表4: モデル(*n*=1, *k*=40)の人手による評価

格助詞候補をひとつだけ生成し、候補をランクしなおすだけで、0.19のBLEUスコア改善がみられている。最後の6列は、*n*と*k*の値をさまざまに変えて実験した結果である。最良の結果は、*n*=20, *k*=10のときに得られた38.91で、BLEUで約1ポイントの改善が見られたが、ここから*n*と*k*を大きくしていっても同様の結果にとどまった。*n*=20, *k*=10のモデルをテストデータに適用した時のBLEUは36.29(ベースラインは35.53)であり、この違いは統計的に有意であった(*p*<.01、ウィルコクソン符号順位検定による)。

4.3 人手による評価

前節のBLEUによる実験結果は、実際にどのような改善を翻訳結果にもたらしているのかを、人手を介した評価で調べてみた。評価方法は、ベストモデル(*n*=20, *k*=10)による出力がベースラインの出力と異なる100文をテストデータから抽出し、2名の被験者にどちらが妥当か、あるいはどちらも同程度妥当(か妥当でない)かを判断してもらった。その際、妥当性は「流暢さ」(fluency)と「適切さ」(adequacy)の2点を別々に判定してもらった。流暢さは正解の翻訳を参照せず2つの出力のみを見ての妥当性、適切さは正解翻訳を考慮に入れての妥当性である。

この評価の結果を表3に示す。表の左半分が流暢さ、右半分が適切さの判定の結果であり、被験者1の結果を縦軸に、被験者1の結果を横軸に示している。S、B、Eはそれぞれ「提案モデルの出力の方が妥当」「ベースラインの出力の方が妥当」「どちらでもない」を表す。表の右半分・左半分それぞれ太字の部分が、2人の被験者が同意した判断を示す。全体的に提案モデルによる解がベースラインよりも妥当と判断されているが、表の結果は流暢さのみにおいて統計的に有意であった。

この評価は*n*=20のモデルの評価であり、この場合、格助詞以外でも異なった翻訳候補が複数、SMTシステムから出力されている。このため、比較する翻訳文の異なり度が大きく、判定が一定しにくい。このような不確定要素を軽減するため、*n*=1, *k*=40のモデルの結果も同じように人手で評価した。ちなみにこのモデルのテストデータでのBLEUは36.09であった。その結果を表4に示す。この実験では流暢さ、適切さともに提案方法において統計的に有意な改善が見られた。なかでも、流暢さにおいては、提案方法の結果のほうがよいと両被験者に判定された文が100文中42文あり、逆に提案手法で悪くなったと判断された文はひとつもなかった。提案手法を保守的に、*n*=1のモデルに使用することで、高い精度で翻訳結果の格助詞を改善することができることを示している。以下、提案手法で改善した翻訳文の例を挙げる。Rが人手による翻訳、Bがベースラインの解、Sが提案モデルによる解である。

R: [検索文字列]にelphrg01と入力し、[検索]をクリックします。
B:検索テキストで、elphrg01入力し[検索]をクリックします。
S:検索テキストで、elphrg01と入力し[検索]をクリックします。

R:返されるエラーの内容については無視してください 。
B:返されるエラーが無視してください。
S:返されるエラーを無視してください。

5 おわりに

本稿では、統計的機械翻訳システムの格助詞生成を改善するモデルを提案した。提案方法を、格助詞候補を拡張したNベスト解に適用することで、非常に高い精度で副作用なく、より適切な格助詞を出力することができた。今後はこのモデルをさらに一般化し、格助詞以外の機能語の生成にも使用して行きたい。

参考文献

[1] 鈴木久美、Kristina Toutanova. 2006. 機械学習による日本語格助詞の予測. 言語処理学会第13回年次大会.

[2] Och, F. J. 2003. Minimum Error-rate Training for Statistical Machine Translation. In *ACL*.

[3] Och, F. J. and H. Ney. 2000. Improved Statistical Alignment Models. In *ACL*.

[4] Och, F. J., D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin and D. Radev. 2004. A Smorgasbord of Features for Statistical Machine Translation. In *NAACL*.

[5] Papineni, K., S. Roukos, T. Ward and W.J. Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *ACL*.

[6] Quirk, C., A. Menezes and C. Cherry. 2005. Dependency Tree Translation: Syntactically Informed Phrasal SMT. In *ACL*.

[7] Suzuki, H. and K. Toutanova. 2006. Learning to Predict Case Markers in Japanese. In *ACL-COLING*.

[8] Toutanova, K., and H. Suzuki. 2007. Generating Case Markers in Machine Translation. To appear in *NAACL-HLT*.