句に基づく機械翻訳のための構文情報を利用したデコーディング

林 健一† 綱川 隆司† 宮尾 祐介† 辻井 潤一†‡§
†東京大学 情報理工学系研究科
‡School of Computer Science, University of Manchester
§National Centre for Text Mining, University of Manchester
{hayashi,tuna,yusuke,tsujii}@is.s.u-tokyo.ac.jp

1　はじめに

統計翻訳モデルは翻訳文のペアを使って学習する数理的なモデルである。近年、多くの研究によって、句に基づく統計翻訳が他の統計翻訳よりも良い翻訳を生成できることが示されてきた [7] [9]。句に基づく統計翻訳では、句を翻訳単位とするが、統計翻訳における句は構文的に意味のある句のことを指すのではなく、単に連続な単語列のことを指す。句に基づく統計翻訳は語順が似た言語間では高精度で翻訳できるが、日本語と英語のように語順が異なる言語間で翻訳を行う場合には構文的に正しくない文を生成してしまうことがある。

近年では構文情報を用いる翻訳として、訓練前・翻訳前に構文情報を使って翻訳元言語の語順を翻訳先言語の語順に近づける手法 [12]、言語モデルの代わりに構文解析モデルを用いる手法 [3]、単語間・句間のアラインメントの代わりに構文木間のアラインメントを用いる手法 [13][4] などが提案されてきた。

本研究では被翻訳文の構文情報を用いてデコーディングを改善する。そして、構文情報を利用して翻訳文の語順を改善し、より構文的に正しい文を生成することを目指す。そのために、被翻訳文の係り受け関係のもとでの、翻訳文の句の順番の条件付確率を推定する。そして、デコーディングにおけるビームサーチで、ある閾値よりもこの条件付確率が小さい仮説を破棄する。条件付確率の推定に必要な係り受け関係は翻訳元言語の構文解析を用いて獲得する。統計翻訳における句は、構文解析器が出力した構文的な句と一致するとは限らないが、構文的な句の情報を統計翻訳の句にマッピングし、それを利用する。

本稿では日英翻訳について取り扱う。日本語と英語は構文構造が異なるため翻訳を行うことが難しいが、より大きい n で n-gram NIST スコアと n-gram BLEU が向上することが実験により確認され、本手法によって句の順番が改善されることが示された。

2　擬似句

統計翻訳における被翻訳文の句にカテゴリーと係り受け関係を付加したものを擬似句と呼ぶことにする。本手法では擬似句のカテゴリーと係り受け関係を得るために構文解析器の出力を利用する。しかし、統計翻訳における句は一般には連続した単語列のことを指し、構文解析器が出力した構文的な句と一致するとは限らない。そのため擬似句を得るためには、構文的な句の情報を統計翻訳における句にマッピングする必要がある。

2.1　擬似句のカテゴリー

まず、擬似句のカテゴリーを得るための手法について説明する。日本語では単語は必ず後ろの単語に係る。そのため、本手法では擬似句の最後の単語に注目し、その単語の品詞を擬似句のカテゴリーとする。図 1 は “同室のものが鍵をとりにきたら、私は午後五時ごろ帰ると伝言してください。” という文の構文解析結果を示したものである。例えば “きたら 、 私” が擬似句の場合、“私” の品詞が名詞なので、この擬似句のカテゴリーを [**名詞**] にする。

図 1: “同室のものが鍵をとりにきたら、私は午後五時ごろ帰ると伝言してください。” の構文解析結果

助詞と名詞が擬似句の最後にある場合、同じ品詞であっても文中における役割が異なることがあるため、カテゴリーを品詞による分類よりもさらに詳細に分ける。[**名詞**] は [**数詞**] と他の [**名詞**] に分割する。助詞は格助詞、接続助詞、副助詞、終助詞という分類があるが、同じ分類でも助詞によって役割が異なるため、助詞ごと

にカテゴリーを分ける。例えば “帰る と 、” のカテゴリーは [**と**]、“が 鍵 を” のカテゴリーは [**を**] とする。“が” については [**が (格助詞)**] と [**が (接続助詞)**] に分類する。ただし終助詞は、どれもほぼ同じ機能を持つので同じカテゴリーに分類する。

接頭語、接尾語、助動詞、判定詞は明確な英語翻訳が存在しないため例外的な語として扱い、本手法では例外的な語以外で最も後ろにある単語を、擬似句のカテゴリーの分類に使用する。図 2 は” 七月十日頃のパリ行きディスカウント航空券が欲しい。” という文の擬似句のカテゴリーである。句の分け方は図の例以外にも多数存在する。四角が擬似句を表し、[•] はカテゴリーを表す。

図 2: 擬似句のカテゴリー

2.2 擬似句の係り受け関係

次に、擬似句の係り受け関係を得るための手法について説明する。構文解析器の出力における係り元の句を P_A、係り先の句を P_B とおく。本手法では擬似句 P_C が P_A の最後の単語を含み、擬似句 P_D が P_B の単語を一つでも含む場合に、擬似句 P_C が P_D に係ると仮定する。

ここでもカテゴリーを得るときと同様に、明確な英語翻訳が存在しない接頭語、接尾語、助動詞、判定詞を例外的な語として扱う。もし P_C が P_A の最後の語を含むがそれが例外的な語であった場合には、P_C が P_D に係るとは仮定しない。P_C が P_A 中の例外的な語以外の最後の単語を含む場合に、P_C が P_D に係ると仮定する。また P_D が例外的な語しか含まない場合にも P_C が P_D に係るとは仮定しない。例えば擬似句 “帰る と” は擬似句 “して (動詞) ください (接尾語)” に係るが、擬似句 “帰る と” は擬似句 “ください (接尾語)” には係らない。図 3 と図 4 は構文解析器が出力した係り受け関係と擬似フレーズ間の係り受け関係の例である。

図 3: 構文解析器が出力した係りうけ関係

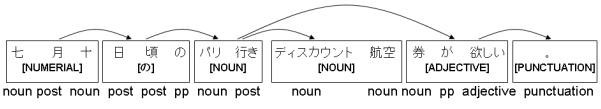

図 4: 擬似フレーズ間の係り受け関係

3 デコーディング

3.1 句の順序の条件付確率の推定

係り受け関係を持つ擬似句の、翻訳後の句の順序を、対訳コーパスを使って推定する手法について説明する。本手法では、対訳コーパス中の被翻訳文の擬似句の係り受け関係の情報が必要となるが、対訳コーパス中の被翻訳文がどのような擬似句に分割され、翻訳されるのかは明確ではない。そのためある条件を満たす全ての擬似句への分割を考慮し、各分割について係り受け関係を考える。その条件は GIZA++[8] から得た単語アラインメントを元に考えるが、本手法で用いるのは両方向のアラインメントの積集合である。擬似句への分割は、被翻訳文だけで考えるのではなく、翻訳ペアを見つけることで行っていく。分割条件は以下の通りである：(i) 翻訳ペアは必ずアラインメントの積集合内の組み合わせを含む、(ii) ある翻訳ペア内の単語と他の翻訳ペア内の単語の組み合わせは、アラインメントの積集合には含まれない。

推定ではまず最初に構文解析器を用いて日本語文の構文解析を行う。次に上記の条件を満たすような全ての擬似句への分割を行う。そして係り受け関係にある擬似句の翻訳後の位置関係に注目し、$count(C_A, C_B, q)$ と $count(order, C_A, C_B, q)$ を得る。C_A と C_B は擬似句 A と B のカテゴリーを指し、A は B に係るとする。また q は被翻訳文が疑問文か叙述文かを示す 2 値変数である。$order$ は句の順序に関する変数で、擬似句 A と翻訳関係にある英語句を C、擬似句 B と翻訳関係にある英語句を D とすると、A と B の句の順序が C と D の順序と同じならば $order$ は $conservation$ に、C と D の順序が反対ならば $order$ は $interchange$ に、A と B が同じ句であるなら $order$ は $same\ phrase$ になる。図 5 は擬似句の翻訳後の順序を示している。

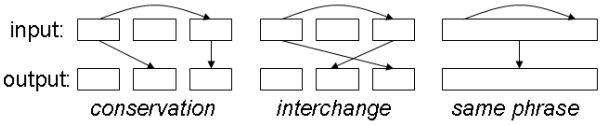

図 5: 擬似句の翻訳後の順序

文 s_i 中の $count(\cdot)$ を $count_i(\cdot)$ と表す。擬似句の分割が一様分布であることを仮定し、以下のように $P(order|C_A, C_B, q)$ を推定する：

$$P(order|C_A, C_B, q) = \frac{\sum_{i \in s_{AB}}^{N} \frac{count_i(order, C_A, C_B, q)}{count_i(C_A, C_B, q)}}{N}。 \quad (1)$$

s_{AB} は C_A と C_B を両方とも含むような文のインデックスの集合、N は s_{AB} の要素数を表す。

3.2 ビームサーチにおける枝刈り

デコーディングにはビームサーチを用いるが、ビームサーチでは同じ単語数を出力した仮説間での相対閾値とビーム幅による枝刈りが行われる [11]。本手法ではこれらに加え条件付確率 $P(order|C_A, C_B, q)$ を使った枝刈りを行う。以下で本手法の手順を示す。

まず、新しい仮説を生成したときに、翻訳する日本語句の擬似句 A を取得する。

そして、A と係り受け関係にある全ての擬似句を得る。その擬似句のうちの一つを B とおく。被翻訳文の翻訳されていない部分の句分割はこの時点で決定していないため、B はすでに翻訳した部分から検索する。A、B の翻訳である英語句をそれぞれ C、D とする。

次に、英語句は文頭から生成されるため D の位置は必ず C よりも前であることと、日本語は必ず後ろに係ることに注意し、以下のように計算する：(i)A が B に係る場合、$P_{AB} = P(interchange|C_A, C_B, q)$、(ii)B が A に係る場合、$P_{AB} = P(conservation|C_B, C_A, q)$。

最後に、$P_{AB} \leq \gamma$ が 1 つ以上の B で満たされる場合に仮説を破棄する (γ はパラメータ)。

4 実験

日英翻訳について実験を行った。日本語の構文解析には KNP[2] を用いた。コーパスは IWSLT 2004 BTEC Corpus[1] を用い、訓練に 20000 文、評価には 500 文を使用した。評価文一つあたりに 15 の参照訳がある。枝刈りに用いるパラメータには $\gamma = 0.3$ を使用した。評価は Moses[6] の初期設定との比較により行った。

4.1 実験結果

元のコーパスの日本語文の単語分割を、KNP の出力に基づいた単語分割に変更してから訓練・評価を行っている。単語分割の変更により、Moses と本手法の両方の翻訳精度が向上している。

表 1: BLEU スコアと NIST スコア

		1-gram	2-gram	3-gram
Moses	NIST	5.1232	6.4272	6.7416
	BLEU	78.38	57.31	42.59
using syntactic information	NIST	5.2068	6.5395	6.8978
	BLEU	77.18	56.35	42.18
		4-gram	5-gram	6-gram
Moses	NIST	6.9414	6.9634	6.9709
	BLEU	31.98	24.43	18.74
using syntactic information	NIST	7.1093	7.1314	7.1408
	BLEU	32.11	24.72	19.28

NIST[5] と BLEU[10] による評価は表 1 のようになった。NIST スコアについては Moses よりも良い精度が得られているが BLEU スコアについては 1-gram、2-gram での精度が悪いために Moses と同程度となっている。これは本手法が、情報量の高い単語列を生成する傾向があることを示している。また、n が大きくなるほど n-gram のスコアが良くなっているのは、長い単語列を見ないと句の順序の変化は考慮できないためと考えられ、これらは本手法を使うことで句の順序を改善することができることを示している。

4.2 エラー解析

以下に翻訳例を示す。日本語文を J、Moses の翻訳文を E_{Moses}、本手法の翻訳文を E_{syn} と表す。

4.2.1 語順の改善

次の例は、句の順序が良くない仮説を破棄することで、本手法の語順が改善されたものである：

J 三 時間 後 に 到着 し ます 。

E_{Moses} three hours after i 'll arrive .

E_{syn} i arrive three hours later .

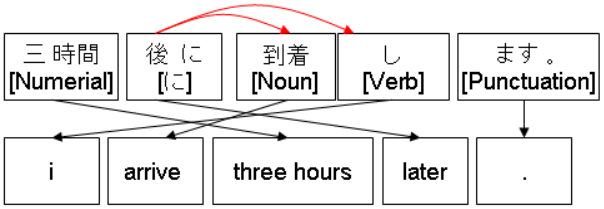

図 6: “三 時間 後 に 到着 し ます 。” の翻訳例

図 6 は “三 時間 後 に 到着 し ます 。” の翻訳を図示したもので、直線が E_{syn} の句間の翻訳関係、曲線が係り受け関係を示している。

$P(conservation|[\textbf{に}], [\textbf{VERB}], declarative) = 0.25$、

$P(conservation|[\textbf{に}], [\textbf{NOUN}], declarative) = 0.15$

が成り立っているため、E_{syn} については”later” が”i” と”arrive” よりも後ろにあり、正しい語順で翻訳できている。

4.2.2 構文的に正しい仮説の保持

構文的に正しい仮説が保持されやすくなることで、語順以外についても生成される文が改善されることがある。

J 今夜 まで に そこ へ 行け ます か 。

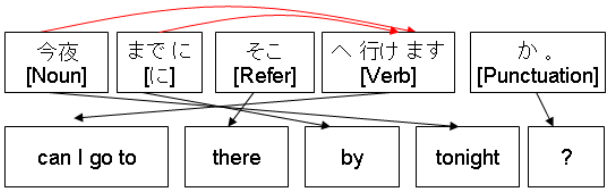

図 7: “今夜 まで に そこ へ 行け ます か 。” の翻訳例

E_{Moses} this evening to get to get ?

E_{Syn} can i go to there by tonight ?

$P(conservation|[\mathbf{NOUN}], [\mathbf{VERB}], interrogative) = 0.29$、

$P(conservation|[\mathbf{に}], [\mathbf{VERB}], interrogative) = 0.26$

この例では語順が改善されているだけでなく、生成された単語自体も良くなっている。これはビームサーチの途中で構文的に正しい仮説だけが保持されたために、単語の生成についても改善したことを示している。

4.2.3 同じ語の生成

次の例で、同じ語を繰り返し生成してしまう傾向が、本手法にあることを示す。この例では E_{syn} 中で “you” が 2 度出力されてしまっている：

J あの 時計 を 見せて くれ ませ ん か 。

E_{Moses} that watch show you ?

E_{syn} that watch will you show me you ?

被翻訳文で近くに存在する単語は似た意味を持つことがある。このような単語が翻訳され同じ単語が生成されたとき、翻訳後にも近くに存在すれば、言語モデルによって誤翻訳を検出することができる。しかし、本手法では、このような単語が比較的に離れた位置に生成される傾向があり、同じ語が繰り返し生成されてしまうことがある。

5 おわりに

本稿では、構文解析器から得られた翻訳元言語の係り受け関係を、統計翻訳における句にマップし、句の順序の条件付確率を計算し、デコーディングのビームサーチで新たに枝刈りを行う手法を提案した。

本手法により、短い単語列よりも長い単語列で翻訳精度が向上していることから、翻訳文の語順が改善されることが示された。また、語順以外でもより良い仮説を保持できることによる改善も見られた。しかし、構文解析が失敗したときに余計な枝刈りを行ってしまう、構文的に同一句内の語順は考慮できない、同じ単語が繰り返し生成されやすいという問題点がある。

より良い語順の翻訳文を生成するためには、より詳細なカテゴリーの分類が必要であると考えている。特に係り先の句では一番後ろの単語に注目するだけでなく、その他の単語に注目したカテゴリー分けが必要である。また今後、句の順序の条件付確率をビームサーチの枝刈りだけに用いるのでなく、仮説のスコア計算に組み合わせることの検討が必要である。

参考文献

[1] IWSLT 2004 BTEC corpus. http://www.slc.atr.jp/IWSLT2004/archives/000211.html.

[2] Sadao Kurohashi And. Kn parser : Japanese dependency/case structure analyzer, 1994.

[3] E. Charniak, K. Knight, and K. Yamada. Syntax-based language models for statistical machine translation, 2003.

[4] Y. Ding and M. Palmer. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proc. ACL*, 2005.

[5] G. Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proc. HLT*, 2002.

[6] P. Koehn, H. Hoang, A. B. Mayne, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. In *ACL, Demonstration Session*, 2007.

[7] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proc. NAACL*, 2003.

[8] F. J. Och and H. Ney. Improved statistical alignment models. In *Proc. ACL*, 2000.

[9] F. J. Och and H. Ney. The alignment template approach to statistical machine translation. *Comput. Linguist.*, 2004.

[10] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation, 2001.

[11] C. Tillmann and H. Ney. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Comput. Linguist.*, 2003.

[12] C. Wang, M. Collins, and P. Koehn. Chinese syntactic reordering for statistical machine translation. In *Proc. EMNLP-CoNLL*, 2007.

[13] Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Proc. ACL*, 2001.