Khafra: 語順並べ替えモデルに対応した 動的計画法に基づく SMT デコーダ

安田 隆浩 *　　越川 満 †　　乾 孝司 †　　山本 幹雄 †

*筑波大学　第三学群情報学類
†筑波大学大学院 システム情報工学研究科

1　はじめに

統計的機械翻訳 (SMT) のデコーダはモデルが与えるスコアを最大化する目的言語文を探索する。しかし探索空間は膨大であるため近似的探索手法を用いる必要がある。フレーズベース SMT の場合、現在はマルチスタックビームサーチ (MB) 法を用いるのが主流である。しかし、ルールベースの機械翻訳システムと比較して 2 桁以上翻訳速度が遅く、実用的なシステムを開発するためには高速なデコーダの研究が必要不可欠である。

試みの一つとして、MB 法と動的計画 (DP) 法を組み合わせた手法が Zens らによって提案されている [Zens and Ney 08]。Zens らの手法では既存の MB 法によるデコーダよりも効率的な pruning による探索空間の限定が可能となり、10 倍程度の高速化を達成している。

しかし、Zens らの論文では語順並べ替えモデルに対応した場合でも高速であることは示されていない。語順並べ替えモデルは日本語から英語への翻訳など語順が大きく入れ替わる言語間での翻訳に有効であることが知られており、代表的なものに Lexicalied Block Orientation(LBO) モデルがある [Tillmann and Zhang 05]。

本稿では、Zens らの手法に LBO モデルを組み込む方法を検討し、組み込んだ場合の速度変化を測定した。結果として、LBO モデルを組み込むと一般の MB デコーダでは翻訳性能は向上する代わりに速度低下が起こっていたが、DP デコーダではほぼ同等の性能向上を達成しながら速度低下は生じないことが明らかになった。

なお、本稿の実験のために開発した DP 法に基づくデコーダは脚注の URL[*1]にて公開している。

2　統計的機械翻訳

2.1　モデルの枠組み

フレーズベース SMT では原言語文 $\boldsymbol{f}$ が与えられたとき、文をフレーズの列 $\boldsymbol{f} = \bar{f}_1, ..., \bar{f}_I$ に分割し、各フレーズ毎に目的言語フレーズ $\bar{e}$ へ翻訳し、並べ替えることによって目的言語文 $\boldsymbol{e} = \bar{e}_1, ..., \bar{e}_I$ を得る。目的言語文 $\boldsymbol{e}$ は対数線形モデルによってスコアが付けられ、スコアが最も良い $\boldsymbol{e}$ が $\boldsymbol{f}$ に対する翻訳文 $\hat{\boldsymbol{e}}$ となる。これは次のような式で表すことができる [Koehn et al. 03]。

$$\hat{\boldsymbol{e}} = \arg_{\boldsymbol{e}} \max_{\boldsymbol{e},\boldsymbol{c}} \sum_j \lambda_j \Phi_j(\boldsymbol{e}, \boldsymbol{c}, \boldsymbol{f}) \tag{1}$$

$\boldsymbol{c} = c_1, .., c_I$ は $\bar{e}_i$ と $\bar{f}_{c_i}$ が翻訳時に対応するフレーズペアであることを示す。Φ_j はスコアを計算する際に用いる素性関数であり、λ_j は Φ_j に対する重みである。素性関数には言語モデル、翻訳モデル、語順並べ替えモデルといった確率モデルの確率値の対数などが用いられる。$\arg_{\boldsymbol{e}} \max_{\boldsymbol{e},\boldsymbol{c}}$ によりスコアが最大となる目的言語文 $\boldsymbol{e}$ を探索するシステムをデコーダと呼ぶ。

2.2　Lexicalized Block Orientation モデル

語順並べ替えモデルで代表的なものが Lexicalized Block Orientation(LBO) モデルである。LBO モデルでは、目的言語側で i 番目および $i+1$ 番目のフレーズ $\bar{e}_i$、$\bar{e}_{i+1}$ に対応する原言語側フレーズ $\bar{f}_{c_i}$ と $\bar{f}_{c_{i+1}}$ との位置関係を、以下のような 3 クラスに分類する [Tillmann and Zhang 05]。

$$class\,(c_i, c_{i+1}) = \begin{cases} mono & (c_{i+1} = c_i + 1 \text{ のとき}), \\ swap & (c_{i+1} = c_i - 1 \text{ のとき}), \\ discont. & (\text{それ以外}). \end{cases} \tag{2}$$

目的言語側で隣接するフレーズ対に対して、*mono* は原言語側において隣接かつ目的言語側と同順であること、*swap* は原言語側において隣接かつ目的言語側と逆順であること、*discont.* は原言語側において両者が離れていることを表す [Tillmann and Zhang 05]。

LBO モデルでは、上記の 3 クラスを用いた各フレーズ対間のリオーダリング確率の積として以下のものを用いる。

$$P(c_1^I|\boldsymbol{e}) \approx \prod_{i=1}^{I} P(class(c_i, c_{i+1})|\bar{e}_i) \tag{3}$$

3　マルチスタックビームサーチデコーダ

MB 法ではフレーズ列に分割した原言語文中からフレーズを選択/翻訳し、翻訳されたフレーズを翻訳した順につなげた部分翻訳を仮説と呼ぶ。ある仮説からさら

[*1] http://www.nlp.mibel.cs.tsukuba.ac.jp/khafra

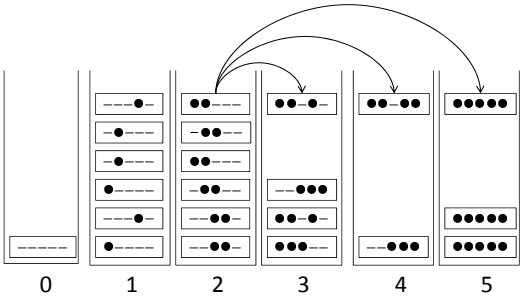

図 1　MB デコーダによる仮説の展開

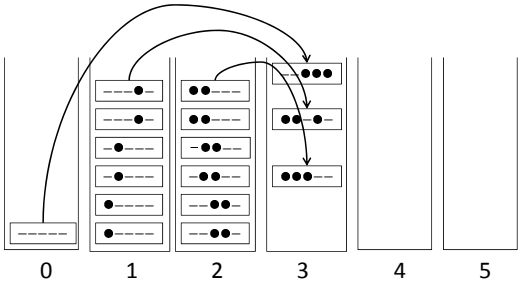

図 2　DP デコーダによる仮説の展開

に別のフレーズを選択/翻訳し、新たな仮説を生成することを仮説の展開と呼ぶ。このときフレーズの翻訳は、フレーズテーブルと呼ばれる原言語と目的言語のフレーズペアからなる集合に存在するものから選択する。原言語フレーズの選択、また原言語フレーズに対する目的言語フレーズの選択の仕方は多数存在するため、ある仮説から展開される仮説の数は多数存在する。その結果、1 つの文を翻訳する際に生成される仮説の数は膨大であるため、これらを効率よく処理する必要がある。

MB デコーダによる仮説の展開を図 1 を用いて説明する。図 1 では原言語文として 5 単語からなる文が与えられたとする。MB デコーダでは仮説は翻訳済み単語数ごとに分けてスタックで管理される。図 1 中の数字はスタックに格納される仮説の翻訳済み単語数を表し、各矩形は仮説を表す。矩形中の “-” は原言語文中の未翻訳単語位置、“●” が翻訳済み単語位置を表す。初期状態では 1 番以降のスタックは空であり、0 番には翻訳済み単語数が 0 である初期仮説のみが格納されている。

仮説の展開は 0 番のスタックから番号順に行う。各スタックでは仮説を 1 つ選択し、その仮説から展開可能な全ての仮説を生成する。生成された仮説は翻訳済み単語数に応じたスタックへ格納する。この処理をスタック中の全ての仮説に対し行った後、次のスタックへ移る。5 番のスタックに格納される仮説では原言語文中の全ての単語が翻訳されたことになり、その中からスコアが最も良い仮説の目的言語文 $\boldsymbol{e}$ を翻訳文 $\hat{\boldsymbol{e}}$ とする。

ＭＢ法では各スタックで管理できる仮説の数を制限し、展開される仮説の数を減らすことで翻訳速度を向上させる。一つのスタックで管理できる仮説の最大数をビーム幅と呼び、余分な仮説を破棄することを pruning と呼ぶ。仮説の数がビーム幅を超える場合には、仮説のスコアによって比較し、ビーム幅分の良い仮説だけを残す [Koehn et al. 03]。

4　動的計画法に基づくデコーダ [Zens and Ney 08]

4.1　DP デコーダ

DP デコーダは、MB デコーダと同様、仮説は翻訳済み単語数が等しいものをひとつのスタックにより管理する。初期状態では 0 番のスタックに初期仮説が格納され、他のスタックは空である。MB デコーダとの最大の違いは仮説の展開の仕方である。MB デコーダでは翻訳済み単語数が同じ仮説に対し、可能な全ての展開を行ったが、DP デコーダでは展開後の仮説の翻訳済み単語数が同じになるように、既存の仮説 (どのスタックでもよい) から展開を行う (図 2)。

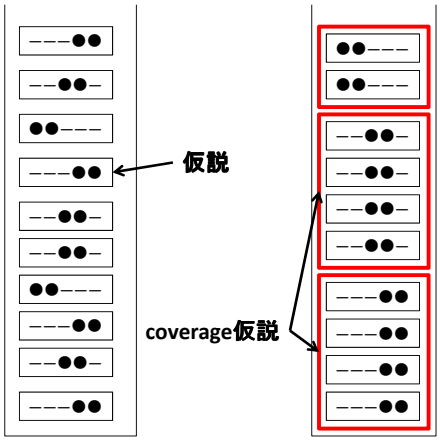

図 3　スタック中の仮説:(左)MB(右)DP

また、新たに coverage 仮説を導入する。coverage 仮説とは原言語文における翻訳済みの単語位置が等しい仮説からなる集合である。DP デコーダではスタック中の仮説は図 3 のように coverage 仮説ごとにまとめられている。仮説の展開の際に原言語文の翻訳済み位置のみに依存する処理があるため、coverage 仮説としてまとめることでそれらの処理を重複して行うことを防ぐことができる。

DP デコーダの詳細を Algorithm1 に示す。line3 で新たな仮説の単語数を決定し、line4 から line10 で展開元の仮説を決定する。この際、まず line5 にて coverage 仮説を決定し、その中から line7 で展開元の仮説を選択する。line11 から line24 で、各モデルのスコアを加算しながら必要に応じて pruning を行い、新たな仮説を生成する。

4.2　early pruning

仮説を実際に生成する前に仮説のスコアを計算し、スタックの閾値と比較する。この結果、閾値を下回るようならこの仮説はスタックに格納される前、生成

される前に破棄される。これを early pruning と呼ぶ [Moore and Quirk 07]。

スタックの閾値は仮説をスコア順に並べた場合にビーム幅との境界に位置する仮説のスコアとなる。仮説のスコアには翻訳モデル (TM)、言語モデル (LM)、語順並べ替えモデル (RM) の確率モデルなどのスコアの合計によって計算されるが、これらスコアは確率値の対数を用いるため加算するごとに単調に減少する。そのため、一部のスコアの合計のみで閾値より小さいことが判明すれば残りを計算する必要は無くなり、その時点でこの仮説を破棄することができる。結果としてスコアの計算量が減少するため、翻訳速度が向上する。

Algorithm1 では LBO モデルを用いない場合、line13、line21 にて early pruning を実行するか判定する。

4.3 スコア順フレーズテーブル

line8 にて line6 で決定した $\bar{f}$ に対する $\bar{e}$ をフレーズテーブルから選択する。このときフレーズテーブルから翻訳モデルスコアが良い順にフレーズを選択することで、スコアが良いものから新しい仮説が生成されやすくする。これにより、閾値が高くなることで early pruning が起こりやすくなる。

また、あるフレーズを用いて仮説の展開を行う際に、展開元の仮説のスコアと翻訳モデルスコアの和がスタックの閾値を下回る場合、残りのフレーズから展開される仮説も閾値を下回ることになる。そのため、残りのフレーズを用いた仮説の展開を行わずに済ますことができる [Zens and Ney 08]。Algorithm1 の line14 における BREAK がこれに当たる。

4.4 coverage pruning

coverage 仮説内の仮説をスコア順に並べることで、展開する仮説を決定する際スコアの良い仮説から展開を行うことができる。このとき、ある仮説のスコアがスタックの閾値を下回る場合、残りの仮説のスコアも閾値を下回ることになる。よって残りの仮説から展開を行わずに済ますことができる。これを coverage pruning と呼ぶ [Zens and Ney 08]。Algorithm1 では line8 がこれに当たる。

5 LBO モデルの組み込み

[Zens and Ney 08] では既存の MB デコーダより高速に探索を行うことができることが示されているが、LBO モデルを用いても高速であることは示されていない。

LBO モデルを用いるうえで重要なのがスコアの計算位置である。モデルの中で最も計算時間が大きいのが言語モデルである。そのため言語モデルの計算は最後にする。

4.3 節の手法を活かす場合、翻訳モデルのスコア計算の前に他のモデルの計算を行うことはできない。よって語順並べ替えモデルの計算は翻訳モデルより後となる。以上から、語順並べ替えモデルは翻訳モデルの後、言語モデルの前に計算するのが良いと考えた。line15 から line18 が語順並べ替えモデルの計算個所である。

Algorithm 1 ＤＰデコーダ

Input: 入力文:$\boldsymbol{f} = f_1^I$

1: スタック $\boldsymbol{s} = s(0), s(1), ..., s(I)$ を初期化
2: 初期仮説 h_{init} を $s(0)$ に追加
3: **for** 展開先スタック $s(i) = s(1)$ to $s(I)$ **do**
4: **for** 新しく翻訳する原言語フレーズ長 $l = 1$ to i **do**
5: **for all** coverage 仮説 cov in $s(i-l)$ **do**
6: **for all** cov における未翻訳原言語フレーズ $\bar{f} : |\bar{f}| = l$ **do**
7: **for all** 仮説 $hypo$incov **do**
8: **if** 4.4 節の coverage pruning の条件を満たす **then**
9: CONTINUE
10: **for all** $\bar{f}$ に対する目的言語フレーズ $\bar{e}$ **do**
11: 翻訳モデルスコア TM を取得する
12: 新しい仮説のスコア $score = hypo$ のスコア $+ TM$
13: **if** 4.2 節の early pruning の条件を満たす **then**
14: BREAK
15: 語順並べ替えモデルスコア RM を計算
16: $score = score + RM$
17: **if** 4.2 節の early pruning の条件を満たす **then**
18: CONTINUE
19: 言語モデルスコア LM を計算
20: $score = score + LM$
21: **if** 4.2 節の early pruning の条件を満たす **then**
22: CONTINUE
23: 新しい仮説 new_hypo を生成
24: new_hypo を $s(i)$ に格納、$s(i)$ 内で coverage 仮説ごとまとめる
25: $s(i)$ の閾値を更新
26: **if** $|s(i)| >$ ビーム幅 **then**
27: pruning $s(i)$
28: $s(I)$ に積まれた仮説のうち最も良いものを出力

6 評価実験

6.1 実験条件

実験には NTCIR-7 特許翻訳タスクで配布された英日対訳コーパスを用いた [Fujii et al. 08]。フレーズテーブルの学習には同学習セットと Moses[Koehn et al. 07] のスクリプトを、各素性関数に対する重みの学習には同 dev セットを使用した。テストセットには NTCIR-7 フォーマルランで配布された文を使用した。

日英方向に翻訳を行い、BLEU を用いて翻訳精度の評価を行った。デコーダには MB デコーダに Moses を使

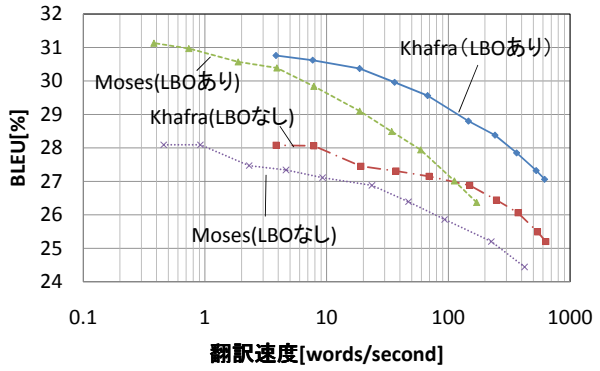

図 4　翻訳速度と BLEU の関係

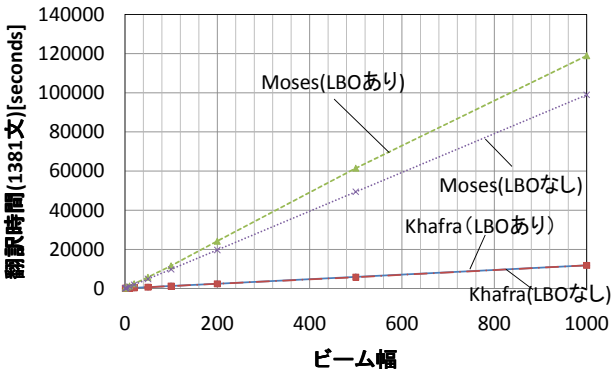

図 5　ビーム幅と翻訳時間の関係

表 1　言語モデルの計算回数 (1381 文)

LBO なし Khafra	LBO あり Khafra
2.97×10^9	2.87×10^9

用し、DP デコーダ (Khafra) を実装し使用した。

6.2　実験結果

Moses および Khafra による翻訳結果の BLEU と翻訳速度の関係を図 4 に示す。図中の各線上にある点は異なるビーム幅のときの性能を表し、右から 1,2,5,10,20,50,100,200,500,1000 の時の翻訳結果となる。翻訳精度が同程度であるときの翻訳速度を比較すると LBO モデルの有無にかかわらず Khafra は Moses に対し、最高約 9 倍の速さで翻訳を行うことが可能であることがわかる。また LBO モデルを用いることで BLEU で 2% から 3% ほど翻訳精度が上昇することがわかる。

図 5 は Moses および Khafra によるテストセット 1381 文の翻訳時間とビーム幅の関係を示したものである。Moses では LBO モデルを用いることで約 2 割以上遅くなるのに対し、Khafra では速度の低下が起こらないことがわかる。

表 1 はテストセット 1381 文をビーム幅 200 で Khafra により翻訳した時の言語モデルの計算回数である。LBO モデルありのほうが言語モデルの計算回数が減少していることがわかる。すなわち、LBO モデルによる pruning の結果達成される言語モデル計算時間の減少が LBO モデルの計算時間の増加分を相殺している。

7　おわりに

本稿では、[Zens and Ney 08] によって提案された動的計画法に基づく SMT デコーダである Khafra を実装し、LBO モデルを対応させた場合でも、既存のマルチスタックビームサーチデコーダより約 9 倍高速に翻訳が行えることを示した。また、MB 法では LBO モデルの導入により顕著な速度低下が生じるが、DP 法ではほぼ速度低下が生じないことが明らかになった。

参考文献

[Koehn et al. 03] P.Koehn, F.J.Och and D.Marcu. 2003. “Statistical phrase-based translation.” In *NAACL ’03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp.48–54.

[Zens and Ney 08] R.Zens and H.Ney. 2008 “Improvements in dynamic programing beam search for phrase-based statistical machine translation.” In *Proceedings of IWSLT*.

[Tillmann and Zhang 05] C.Tillmann and T.Zhang. 2005 “A localized prediction model for statistical machine translation.” In *Proceedings of the 43rd Annual Meeting of the Association for Computational Lingustics*, pp.557–564.

[Fujii et al. 08] A.Fujii, M.Utiyama, M.Yamamoto, T.Utsuro. 2008 “Overview of the Patent Translation Task at the NTCIR-7 Workshop.” In *Proceedings of the 7th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access*. pp.389–400.

[Moore and Quirk 07] R.C.Moore and C.Quirk. 2007 “Faster beam-search decoding for phrasal statistical machine translation.” In *MT Summit XI*.

[Koehn et al. 07] P.Koehn and et al. 2007 “Moses: Open source toolkit for statistical machine translation.” In *Proceedings of the ACL Demo and Poster Session*, pp.177–180.