

18. デコーダ概要

内山将夫@NICT
mutiyama@nict.go.jp

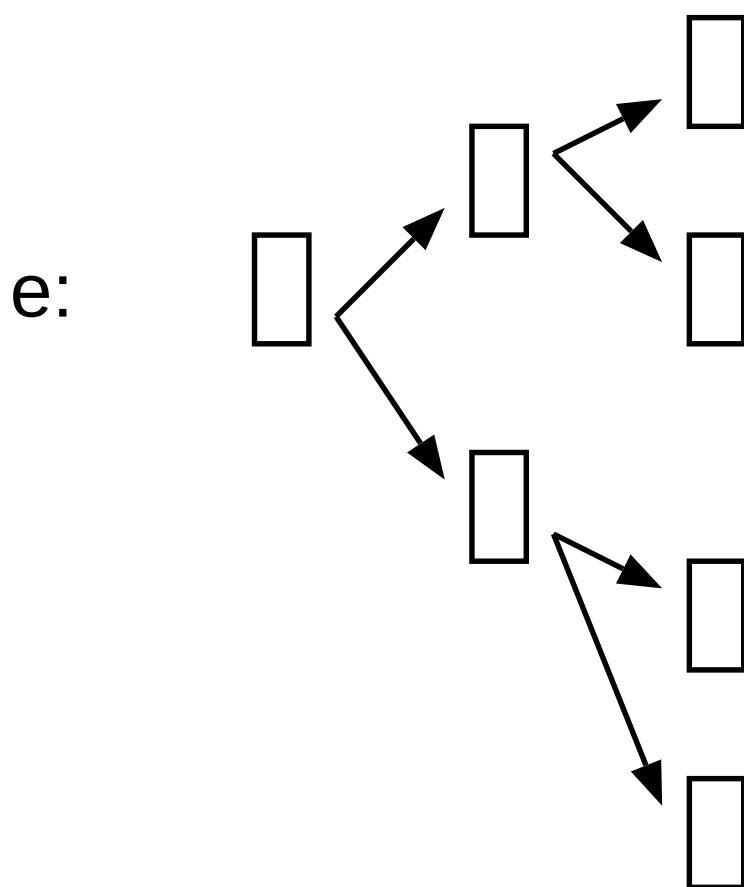
デコーダの役割

$$\hat{\mathbf{e}} = \arg \max_{\mathbf{e}} \sum_i \lambda_i h_i(\mathbf{e}, \mathbf{f})$$

における $\hat{\mathbf{e}}$ の探索.

$\mathbf{f}$ を部分的に訳していったって、いくつかの仮説を作り、そこから最適なものを得る.

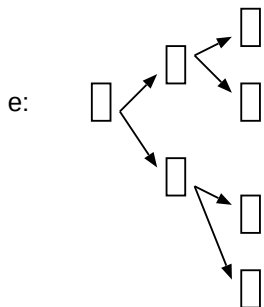
$\mathbf{f}$:



デコーダの基本要素

仮説 (部分翻訳) を生成し，不要なものを消し，良いものを残す.

- 仮説の生成 (部分翻訳の生成)

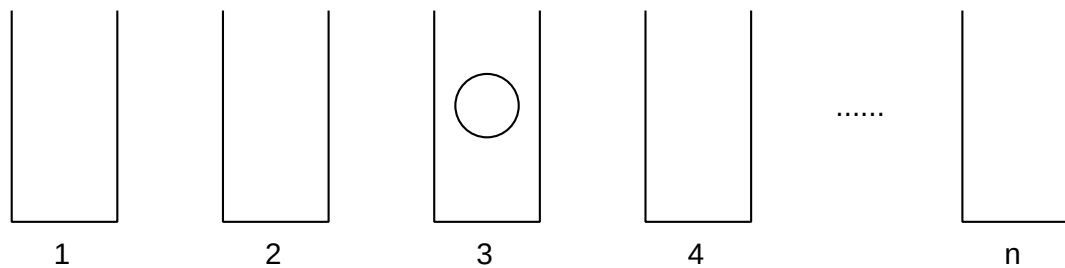


- 仮説の枝刈り (不要な仮説を消す)
- 最も良い仮説の探索

Multi-Stack Beam-Search

$$\mathbf{f} = f_1 f_2 f_3 \dots f_n$$

とすると，これまでに翻訳された $\mathbf{f}$ の単語数に応じて n 個のスタックをつくる．



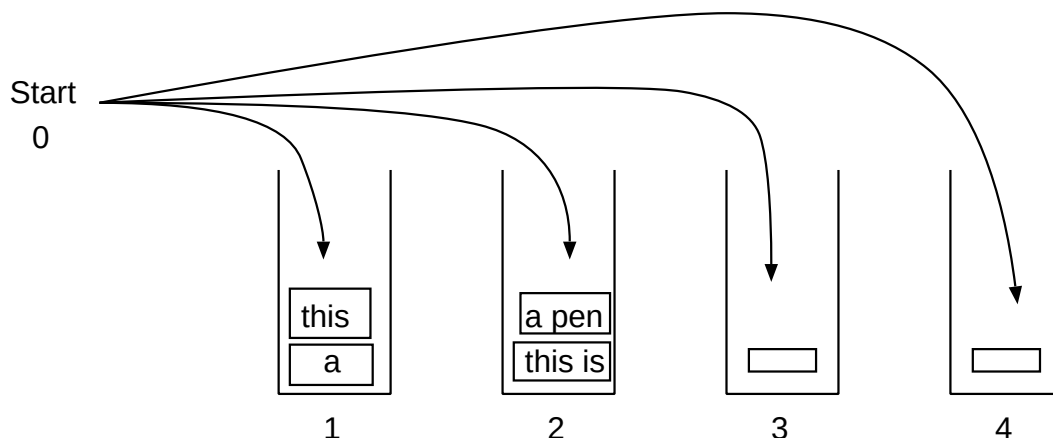
3番目のスタックには， $\mathbf{f}$ の単語を 3 つ訳した結果の部分翻訳を示す仮説が積まれる．たとえば，

$$\mathbf{f} = f_1 \overline{f_2} f_3 \overline{f_4} \overline{f_5} f_6$$

では， f_2 , f_4 , f_5 を翻訳した部分に相当する仮説が積まれる．

初期仮説の積み方

初期仮説 (カバー0) から, 最初の 1 つのフレーズ翻訳によりカバーされる f の側の単語数に応じて仮説をスタックに分配する.



たとえば, 「this is a pen .」を翻訳するときに, フレーズの候補として

- 「this」 → 「これ」「ここ」...
- 「this is」 → 「これは」「それを」...
- 「a」 → 「した」「では」...
- 「a pen」 → 「、ペン」...

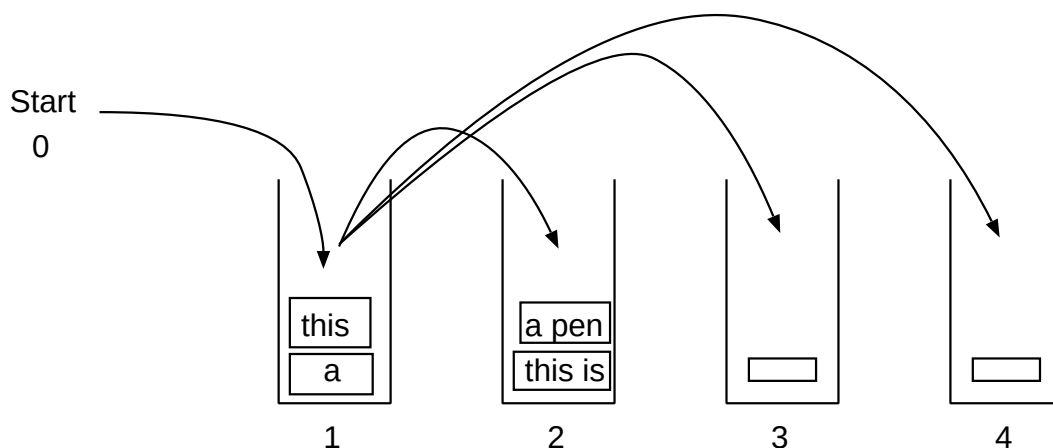
などがある. このとき,

- 「this」と「a」は, カバー1のスタックに載り
- 「this is」と「a pen」は, カバー2のスタックに載る

その他, 全てのフレーズについて, そのフレーズがカバーする単語数に応じて, スタックに分配する.

カバー1のスタックからの仮説の展開

カバー1のスタック中の各仮説について、その仮説に後続可能なフレーズを翻訳して、その結果としてカバーした単語数に相当するスタックに載せる。



たとえば、「this」のあとに

- 「is」を翻訳すると、その結果はカバー2のスタックに
- 「a」を翻訳すると、その結果はカバー2のスタックに
- 「a pen」を翻訳すると、カバー3のスタックに

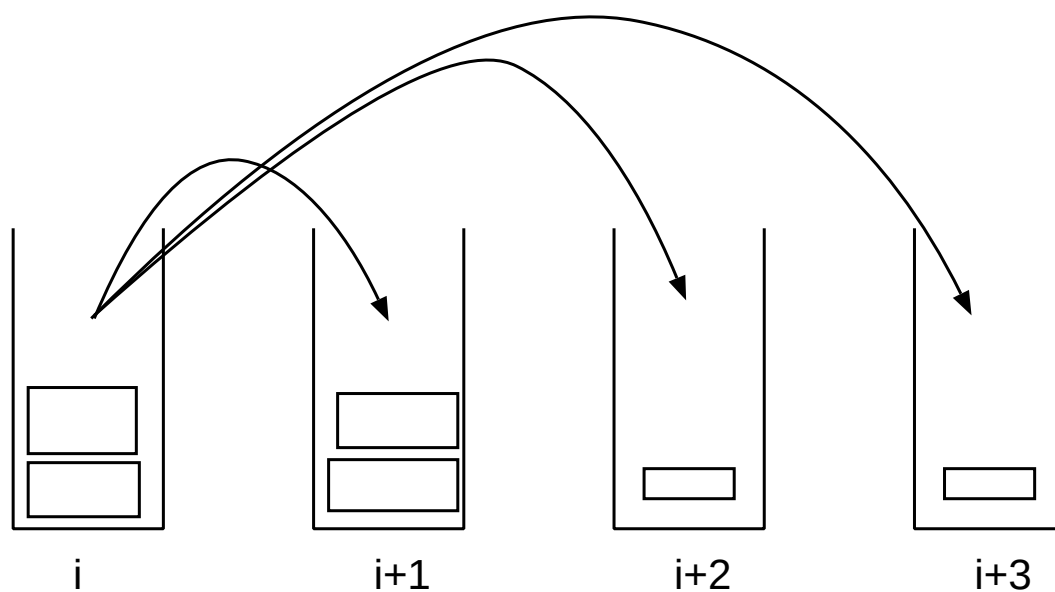
積まれる。その結果として、それぞれ、「this is」「this a」「this a pen」とその訳が積まれている。また、「a」のあとに

- 「this」を翻訳すると、カバー2のスタックに
- 「is」を翻訳すると、カバー2のスタックに
- 「this is」を翻訳すると、カバー3のスタックに

積まれる。その結果として、それぞれ、「a this」「a is」「a this is」とその訳が積まれている。

カバー 2,3,4... i と進む

カバー i のスタックにおいて，入力文から，次の未翻訳のフレーズを翻訳し，そのフレーズを，カバー i のスタック中の部分翻訳に追加した結果の翻訳を，その翻訳がカバーする単語数に対応するスタックに積む．



以上を $1 \leq i \leq n$ まで繰り返すと，スタック n を処理したときには， $f_1, f_2, \dots, f_n$ の全ての単語が翻訳されている．

枝刈りの必要性

1 単語は 1 単語にしか訳されないとしても，これまでの方法では，

- 最初のスタックに n 単語
- 次のスタックに $n - 1$ 単語

...

となって， $n!$ の仮説ができる．フレーズを考えると多くなる．

→ 仮説の枝刈りをしないといけない．

スタックに積む仮説：翻訳オプション

- 仮説のそれぞれは，それぞれが1つの F のフレーズ $\bar{f}$ をもつ
- このフレーズに対応する E のフレーズ $\bar{e}$ ももつ
- 翻訳オプションは，以下の4つの要素を持つ
 - $\bar{f}$ の f 中での開始位置
 - $\bar{f}$ の f 中での終了位置
 - $\bar{e}$
 - フレーズを翻訳するときのコスト

$$= -\sum \lambda_i f_i(\bar{e}, \bar{f})$$

フレーズテーブルにあるフレーズスコアの重み付きの和 $\times -1$

翻訳オプションは， f 中の全てのフレーズについて，あらかじめ作っておく．

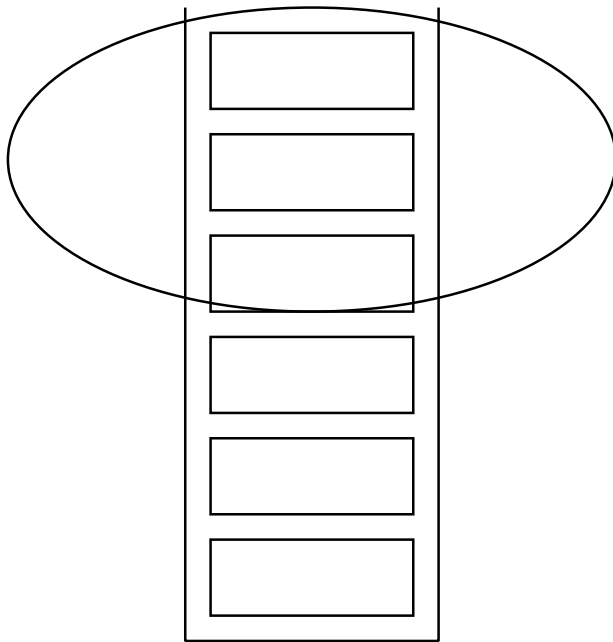
仮説の構成物

- 翻訳オプション
- 1つ前の仮説へのリンク. (これを辿っていけば, どのフレーズがどのフレーズに訳されたかがわかるため, 翻訳文を生成できる)
- これまでにカバーされた f の単語の位置 (重複して同一の単語を訳すことを防ぐため)
- 2つ前までの e の単語. (3-gram 言語モデルの計算に必要)
- これまでのコストの和
- 予測コスト

コスト最小の仮説を探す.

仮説の枝刈り (Beam Search)

予測全体コストが小さい仮説のみを選んで、
コストが大きいものは捨てる



予測全体コスト
= これまでのコスト
+ これからの予測コスト

翻訳した f の数が等しい仮説が載る

これまでのコスト

= Σ フレーズの翻訳コスト

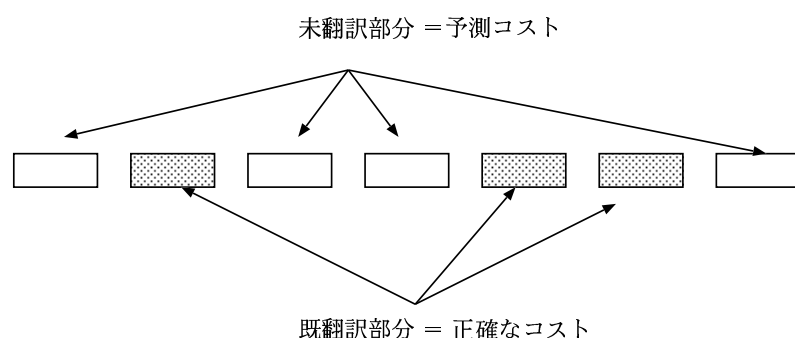
+ 翻訳された e の言語モデルからのコスト

+ 語順のコスト

翻訳された e の言語モデルからのコスト

$$= -\Sigma \log P(e_i | e_{i-2}, e_{i-1})$$

予測全体コスト = 既翻訳のコスト + 未翻訳のコスト



既翻訳部分のコスト = 正確なコスト
= Σ フレーズの翻訳コスト
+ 翻訳文の言語モデルコスト
+ 語順のコスト

未翻訳部分のコスト = 予測コスト
= Σ フレーズの翻訳コスト
+ Σ フレーズ内言語モデルコスト

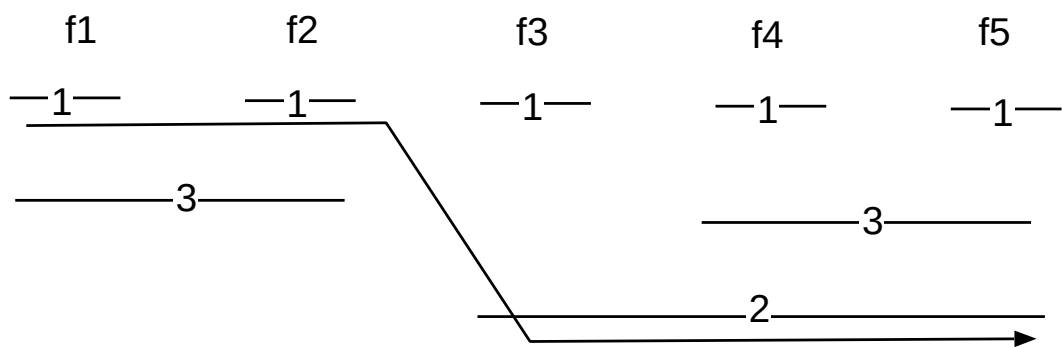
予測コストにおいては，未翻訳のフレーズの翻訳コストの和と未翻訳のフレーズのフレーズ内言語モデルのコストの和を利用する．

フレーズの翻訳コストは，フレーズテーブルにリストされているスコアの和に -1 を掛けたものである．

フレーズ内言語モデルコストは， $\bar{e} = e_1e_2...e_n$ についての言語モデルによる確率の対数に -1 を掛けたものである．

フレーズ内言語モデルコストは，フレーズ間の繋ぎ目をまたいだ確率の計算はしない．

予測コストの計算

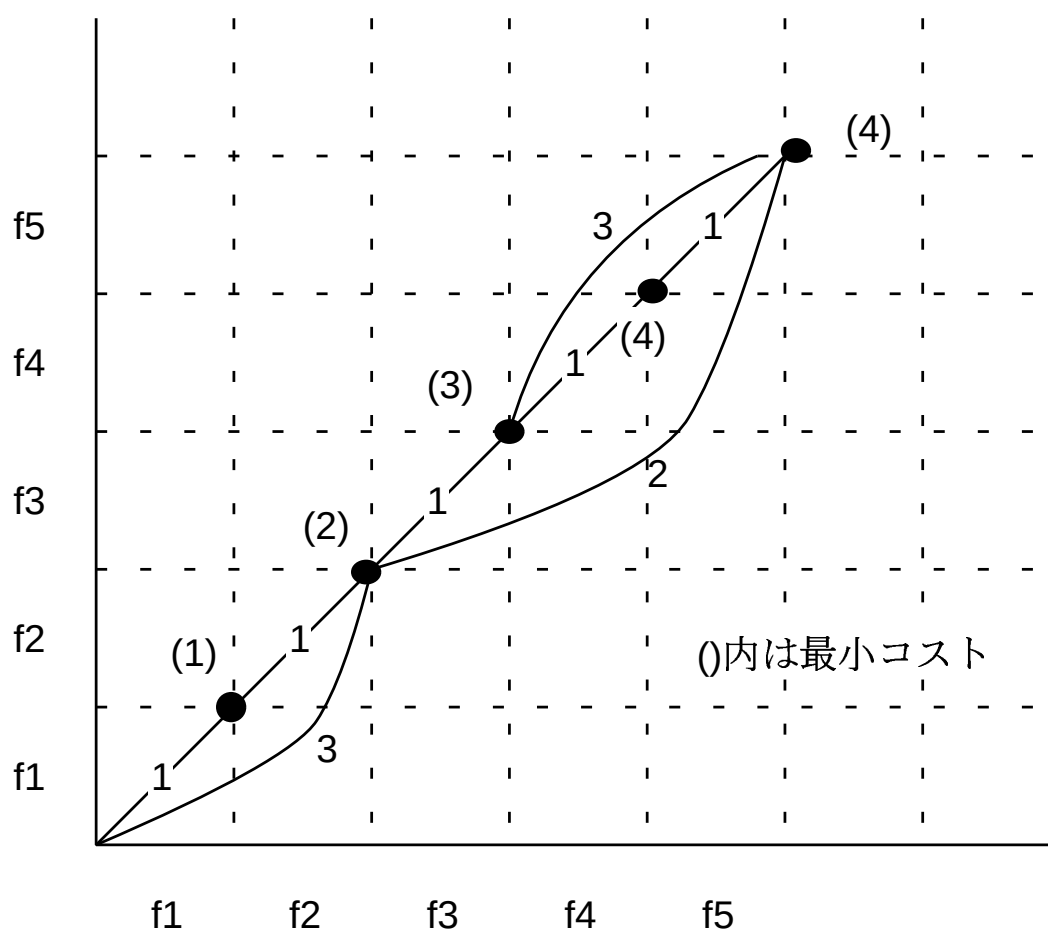


最小コストパスを通ることにより，翻訳コストを計算する

各フレーズのコストは，そのフレーズに対応する，複数の翻訳の中から最小のコストのものを選ぶ．

したがって，予測コストは，最小のコストとなる．

DPを利用した最小コストパスの計算法



こちらに動きながら、翻訳オプションのあるものについてエッジをたどる

実行例

- 入力: this is a pen .
- 出力: これは 、 ペン である 。

翻訳オプション

this ここ, その, て この, この, 上記, ...
is され て, に, も, れる, の は, に は, ...
a では、, (, し た, A, a, ...
pen 筆, ペン の, PEN, 入力 ペン, pen, ペン で, ...
. なる。., される。.,)。., し ている。., れる。., ...

this is を, すなわち, は その, それ を, これ が, この ことは, ...
is a がある, が、, と は、, は その, が, の は、, を、
a pen、ペン, a pen ., た ペン, と、ペン, の ペン, ...
pen . ペン である。

this is a が, この ことが, この ことは、, これ が, これ が、, ...

is a pen ϕ
a pen . ϕ
this is a pen ϕ
is a pen . ϕ
this is a pen . ϕ

[入力フレーズ;ID]

翻訳フレーズ<ID>, pC=対数フレーズ翻訳確率, c=コスト

[this;10]

ここ<1341>, pC=-1.35766, c=-2.1591
その<1648>, pC=-1.04392, c=-1.67822
て この<1845>, pC=-1.27158, c=-2.15583
この<1364>, pC=-0.416247, c=-1.06726
上記<2943>, pC=-1.61984, c=-2.27863
の この<2235>, pC=-1.53633, c=-2.30776
これ<1394>, pC=-0.817653, c=-1.53584
本<3271>, pC=-0.766849, c=-1.37588
これら<1402>, pC=-1.6298, c=-2.3598
以上<2691>, pC=-1.69762, c=-2.21865
これ に<1396>, pC=-1.50672, c=-2.11998
に この<2110>, pC=-1.63886, c=-2.28144
で<239>, pC=-2.04413, c=-2.26692
は<2315>, pC=-1.93096, c=-2.11451
で この<1944>, pC=-1.7142, c=-2.32927
この よう<1373>, pC=-1.35286, c=-2.27915
以上 の<2692>, pC=-1.91313, c=-2.27875
それ<1659>, pC=-1.46538, c=-2.1846
該<2747>, pC=-1.52524, c=-2.26044
当該<3184>, pC=-1.59398, c=-2.34628

[is;5]

されて<3421>, pC=-2.11081, c=-2.02628
 に<299>, pC=-1.48134, c=-1.66314
 も<344>, pC=-1.47625, c=-1.85074
 れる<400>, pC=-1.29528, c=-1.88471
 のは<2251>, pC=-1.41678, c=-2.04515
 には<2133>, pC=-1.43534, c=-1.75688
 とは<2002>, pC=-1.72335, c=-2.04814
 が<1325>, pC=-0.470061, c=-0.647842
 のが<2234>, pC=-1.40044, c=-2.0277
 がある<1328>, pC=-1.52732, c=-1.96323
 れ<2527>, pC=-1.38216, c=-1.94951
 の<2208>, pC=-1.76538, c=-1.90982
 を<417>, pC=-0.577587, c=-0.729719
) を<873>, pC=-1.77558, c=-1.93028
) が<847>, pC=-1.68848, c=-1.89245
) は<870>, pC=-1.77264, c=-2.02156
 される<154>, pC=-1.8255, c=-2.03365
 では<1951>, pC=-1.51902, c=-1.78469
 され<3419>, pC=-1.86897, c=-2.05372
 は<2315>, pC=-0.542636, c=-0.726186

[a;3]

では、<1952>, pC=-1.99372, c=-1.94916
 (<747>, pC=-1.50532, c=-1.732
 した<1473>, pC=-1.4491, c=-1.74456
 A<1049>, pC=-0.853386, c=-1.32077
 a<1121>, pC=-1.09356, c=-1.55888
 には、<2134>, pC=-1.73294, c=-1.75383
 、<8>, pC=-0.684518, c=-0.886027
 は<2315>, pC=-0.824683, c=-1.00823
 は、<2316>, pC=-1.20584, c=-1.38524
 れた<2531>, pC=-1.49016, c=-1.97397
 た<1666>, pC=-0.745912, c=-1.23491
 で、<1899>, pC=-1.57655, c=-1.89724
 な<2033>, pC=-1.18053, c=-1.66961
 する<174>, pC=-1.01081, c=-1.39731
 された<1431>, pC=-1.85698, c=-1.80738
 て、<1786>, pC=-1.21214, c=-1.82842
 では<1951>, pC=-1.64188, c=-1.90755
 ある<96>, pC=-1.07434, c=-1.72888
 には<2133>, pC=-1.46608, c=-1.78763
 である<241>, pC=-1.53484, c=-1.77665

[pen;7]

筆<572>, pC=-1.09842, c=-1.91969
 ペンの<4625>, pC=-1.00945, c=-1.76537
 PEN<4601>, pC=-1.16262, c=-1.98389

入力 ペン<4635>, pC=-1.60213, c=-2.38855
pen<4603>, pC=-1.41516, c=-2.23643
ペン で<570>, pC=-1.23726, c=-2.03232
ペン で ある<4623>, pC=-1.60909, c=-2.33354
ペン 筆跡<4628>, pC=-1.47485, c=-2.45062
て ペン<4607>, pC=-0.999509, c=-2.1773
ペン に<4624>, pC=-1.4014, c=-2.30551
で ペン<4609>, pC=-1.37715, c=-2.31839
ペン 入力<4627>, pC=-1.76514, c=-2.49763
に ペン<4611>, pC=-1.3145, c=-2.23882
の ペン<566>, pC=-1.04526, c=-1.9127
入力<4634>, pC=-2.10195, c=-2.58898
ペン 2 の<4620>, pC=-1.61217, c=-2.45588
スタイラス<4614>, pC=-1.67264, c=-2.49391
タッチ<4615>, pC=-1.79274, c=-2.58717
タッチ ペン<4616>, pC=-1.72678, c=-2.5611
ペン<569>, pC=-0.267554, c=-1.04067

[.;1]

なる 。<294>, pC=-1.10377, c=-1.88984
される 。<155>, pC=-1.40193, c=-1.30851
) 。<33>, pC=-1.47075, c=-1.6819
し て いる 。<167>, pC=-1.67653, c=-1.44244
れる 。<401>, pC=-0.996328, c=-1.63918
さ れ て いる 。<150>, pC=-1.91711, c=-1.17936
ある 。<97>, pC=-0.907851, c=-1.72634
する 。<175>, pC=-1.14859, c=-1.62012
て いる 。<208>, pC=-0.942884, c=-1.23263
。<10>, pC=-0.24766, c=-0.713159
も の で ある 。<349>, pC=-1.90485, c=-1.65041
な っ て いる 。<291>, pC=-1.63915, c=-1.49152
と なる 。<275>, pC=-1.72258, c=-1.72477
れ て いる 。<392>, pC=-1.52524, c=-1.34756
よ う に な っ て いる 。<368>, pC=-2.508, c=-1.76419
に な っ て いる 。<309>, pC=-2.13272, c=-1.64248
に なる 。<310>, pC=-1.61102, c=-1.75654
と な っ て いる 。<274>, pC=-2.35949, c=-1.75109
ら れ て いる 。<382>, pC=-2.07856, c=-1.88984
で ある 。<242>, pC=-1.23406, c=-1.14991

[this is;9]

を<417>, pC=-1.54862, c=-1.70076
すなわち<1528>, pC=-1.73591, c=-2.31995
は その<2385>, pC=-1.71699, c=-2.20306
それ を<4688>, pC=-1.64736, c=-2.29564
これ が<4642>, pC=-0.911397, c=-1.62708
この こと は<4666>, pC=-1.43752, c=-2.30474

は これ<2383>, pC=-1.71766, c=-2.34747
 ここで は<1345>, pC=-1.89881, c=-2.25273
 が<1325>, pC=-1.50635, c=-1.68413
 という の は<4691>, pC=-1.93868, c=-2.18854
 それ は<4687>, pC=-1.37946, c=-2.20885
 が この<3376>, pC=-1.57157, c=-2.13519
 が これ<4657>, pC=-1.5425, c=-2.23305
 が これ に<4658>, pC=-1.73319, c=-2.17005
 が これ は<4659>, pC=-1.57178, c=-2.24404
 は<2315>, pC=-1.45277, c=-1.63632
 以上 が<4652>, pC=-1.71494, c=-2.34458
 これ を<1401>, pC=-0.962022, c=-1.51385
 て これ を<4690>, pC=-1.80236, c=-2.40022
 これ は<1399>, pC=-0.652242, c=-1.35359

[is a;4]

が ある<1328>, pC=-1.67738, c=-2.1133
 が 、<1326>, pC=-1.28772, c=-1.62281
 と は 、<2003>, pC=-1.77381, c=-1.85223
 は その<2385>, pC=-1.62439, c=-2.11046
 が<1325>, pC=-0.863859, c=-1.04164
 の は 、<2252>, pC=-1.68919, c=-2.00883
 を 、<2587>, pC=-1.52488, c=-1.88536
 は<2315>, pC=-0.660629, c=-0.84418
 を<417>, pC=-1.0029, c=-1.15503
) は<870>, pC=-1.94566, c=-2.19457
) は 、<871>, pC=-1.88151, c=-1.8389
 の<2208>, pC=-1.9946, c=-2.13903
 に は 、<2134>, pC=-1.76211, c=-1.783
 で は<1951>, pC=-1.89019, c=-2.15586
 に は<2133>, pC=-1.62461, c=-1.94616
 で は 、<1952>, pC=-2.01248, c=-1.96793
 と は<2002>, pC=-1.87138, c=-2.19618
 は 、 この<2324>, pC=-1.91441, c=-2.1825
 は 、<2316>, pC=-0.796328, c=-0.975726
 に<299>, pC=-1.64566, c=-1.82745

[a pen;2]

、 ペン<562>, pC=-0.753549, c=-1.62573
 a p e n . <563>, pC=-2.12552, c=-3.23558
 た ペン<564>, pC=-0.932183, c=-2.08682
 と 、 ペン<565>, pC=-1.4245, c=-2.38424
 の ペン<566>, pC=-0.944163, c=-1.8116
 は 、 ペン<567>, pC=-1.31232, c=-2.19307
 は ペン<568>, pC=-1.16434, c=-2.07313
 ペン<569>, pC=-0.44751, c=-1.22063
 ペン で<570>, pC=-1.223, c=-2.01806

握り<571>, pC=-1.77062, c=-2.59188
筆<572>, pC=-1.377, c=-2.19827
例えば ペン 入力<573>, pC=-1.84151, c=-3.13061

[pen .;6]

ペン である 。<4599>, pC=-0.68607, c=-1.08455

[this is a;8]

が<1325>, pC=-1.70286, c=-1.88064
この こと が<4640>, pC=-1.41113, c=-2.60653
この こと は 、<4641>, pC=-1.55813, c=-2.26904
これ が<4642>, pC=-1.07722, c=-1.7929
これ が 、<4643>, pC=-1.31737, c=-1.97721
これ で は 、<4644>, pC=-1.62747, c=-1.9997
これ に<1396>, pC=-1.88038, c=-2.49364
これ は<1399>, pC=-0.923228, c=-1.62458
これ は 、<4645>, pC=-1.03133, c=-1.43306
これ は 、 いわゆる<4646>, pC=-1.85256, c=-2.76921
これ は , <4647>, pC=-1.81499, c=-2.83232
以上 が<4652>, pC=-1.67579, c=-2.30544
これ は また 、<4649>, pC=-1.71063, c=-2.51877
これら が<4650>, pC=-1.64068, c=-2.54958
すなわち<1528>, pC=-1.87628, c=-2.46032
ただし 、 これ は<4651>, pC=-1.78216, c=-2.85429
の<2208>, pC=-2.20365, c=-2.34808
は<2315>, pC=-1.79407, c=-1.97762
は 、<2316>, pC=-1.87083, c=-2.05023
も<344>, pC=-1.92003, c=-2.29452

[is a pen;0]

[a pen .;0]

[this is a pen;0]

[is a pen .;0]

[this is a pen .;0]

予測コストの計算

```
4 . |
3 pen |
2 a |
1 is |
0 this |
+-----+-----+-----+-----+-----+
0 this | 1 is | 2 a | 3 pen | 4 .
```

computing future cost from 0 to 0

```
can get to 1 with cost -2.1591
can get to 1 with cost -1.67822
can get to 1 with cost -1.06726
=> cheapest way: -1.06726
```

this
この

computing future cost from 0 to 1

```
can get to 1 with cost -2.1591
can get to 1 with cost -1.67822
can get to 1 with cost -1.06726
can get to 2 with cost -1.70076
can get to 2 with cost -1.62708
can get to 2 with cost -1.51385
can get to 2 with cost -1.35359
=> cheapest way: -1.35359
```

this is
これは

computing future cost from 0 to 2

```
can get to 1 with cost -2.1591
can get to 1 with cost -1.67822
can get to 1 with cost -1.06726
can get to 2 with cost -1.70076
can get to 2 with cost -1.62708
can get to 2 with cost -1.51385
can get to 2 with cost -1.35359
can get to 3 with cost -1.88064
can get to 3 with cost -1.7929
can get to 3 with cost -1.62458
can get to 3 with cost -1.43306
=> cheapest way: -1.43306
```

this is a
これは、

computing future cost from 0 to 3

```
can get to 1 with cost -2.1591
can get to 1 with cost -1.67822
can get to 1 with cost -1.06726
can get to 2 with cost -1.70076
can get to 2 with cost -1.62708
can get to 2 with cost -1.51385
can get to 2 with cost -1.35359
can get to 3 with cost -1.88064
can get to 3 with cost -1.7929
```

```

    can get to 3 with cost -1.62458
    can get to 3 with cost -1.43306
    can get to 4 with cost -2.97932
    can get to 4 with cost -2.57422
    can get to 4 with cost -2.47373
    => cheapest way: -2.47373
computing future cost from 0 to 4
    can get to 1 with cost -2.1591
    can get to 1 with cost -1.67822
    can get to 1 with cost -1.06726
    can get to 2 with cost -1.70076
    can get to 2 with cost -1.62708
    can get to 2 with cost -1.51385
    can get to 2 with cost -1.35359
    can get to 3 with cost -1.88064
    can get to 3 with cost -1.7929
    can get to 3 with cost -1.62458
    can get to 3 with cost -1.43306
    can get to 4 with cost -2.97932
    can get to 4 with cost -2.57422
    can get to 4 with cost -2.47373
    can get to 5 with cost -2.51762
    => cheapest way: -2.51762
computing future cost from 1 to 1
    can get to 1 with cost -2.02628
    can get to 1 with cost -1.66314
    can get to 1 with cost -0.647842
    => cheapest way: -0.647842
computing future cost from 1 to 2
    can get to 1 with cost -2.02628
    can get to 1 with cost -1.66314
    can get to 1 with cost -0.647842
    can get to 2 with cost -2.1133
    can get to 2 with cost -1.62281
    can get to 2 with cost -1.04164
    can get to 2 with cost -0.84418
    => cheapest way: -0.84418
computing future cost from 1 to 3
    can get to 1 with cost -2.02628
    can get to 1 with cost -1.66314
    can get to 1 with cost -0.647842
    can get to 2 with cost -2.1133
    can get to 2 with cost -1.62281
    can get to 2 with cost -1.04164
    can get to 2 with cost -0.84418
    can get to 3 with cost -2.27357
    can get to 3 with cost -1.86847

```

-1.43306 -1.04067
 this is a + pen
 これは、+ ペン

-1.43306 -1.08455
 this is a + pen .
 これは、+ ペン である。

is
 が

is a
 は

-0.84418 -1.04067
 is a + pen

```

=> cheapest way: -1.86847
computing future cost from 1 to 4
  can get to 1 with cost -2.02628
  can get to 1 with cost -1.66314
  can get to 1 with cost -0.647842
  can get to 2 with cost -2.1133
  can get to 2 with cost -1.62281
  can get to 2 with cost -1.04164
  can get to 2 with cost -0.84418
  can get to 3 with cost -2.27357
  can get to 3 with cost -1.86847
  can get to 4 with cost -1.92873
=> cheapest way: -1.92873
computing future cost from 2 to 2
  can get to 1 with cost -1.94916
  can get to 1 with cost -1.732
  can get to 1 with cost -1.32077
  can get to 1 with cost -0.886027
=> cheapest way: -0.886027
computing future cost from 2 to 3
  can get to 1 with cost -1.94916
  can get to 1 with cost -1.732
  can get to 1 with cost -1.32077
  can get to 1 with cost -0.886027
  can get to 2 with cost -1.62573
  can get to 2 with cost -1.22063
=> cheapest way: -1.22063
computing future cost from 2 to 4
  can get to 1 with cost -1.94916
  can get to 1 with cost -1.732
  can get to 1 with cost -1.32077
  can get to 1 with cost -0.886027
  can get to 2 with cost -1.62573
  can get to 2 with cost -1.22063
  can get to 3 with cost -1.97058
  can get to 3 with cost -1.93378
=> cheapest way: -1.93378
computing future cost from 3 to 3
  can get to 1 with cost -1.91969
  can get to 1 with cost -1.76537
  can get to 1 with cost -1.04067
=> cheapest way: -1.04067
computing future cost from 3 to 4
  can get to 1 with cost -1.91969
  can get to 1 with cost -1.76537
  can get to 1 with cost -1.04067
  can get to 2 with cost -1.08455

```

は + ペン
 -0.84418 -1.08455
 is a + pen .
 は + ペン である 。
 a
 、
 a pen
 ペン
 (微妙に数値が違う)
 一つ前のフレーズを考慮して
 対象言語の言語モデルのコストを
 計算していると思われる。
 -0.886027 -1.08455
 a + pen .
 、 + ペン である 。
 pen
 ペン
 pen .

```

=> cheapest way: -1.08455
computing future cost from 4 to 4
    can get to 1 with cost -1.88984
    can get to 1 with cost -1.30851
    can get to 1 with cost -1.17936
    can get to 1 with cost -0.713159
=> cheapest way: -0.713159
future costs from 0 to 0 is -1.06726
future costs from 0 to 1 is -1.35359
future costs from 0 to 2 is -1.43306
future costs from 0 to 3 is -2.47373
future costs from 0 to 4 is -2.51762
future costs from 1 to 1 is -0.647842
future costs from 1 to 2 is -0.84418
future costs from 1 to 3 is -1.86847
future costs from 1 to 4 is -1.92873
future costs from 2 to 2 is -0.886027
future costs from 2 to 3 is -1.22063
future costs from 2 to 4 is -1.93378
future costs from 3 to 3 is -1.04067
future costs from 3 to 4 is -1.08455
future costs from 4 to 4 is -0.713159

```

ペン である。

.

。

仮説の探索

```
0[] +- 1[this=>ここ] --- 174[is=>されて] ...
    +- 2[this=>その]
    +- ...
    +- 21[is=>されて]
    +- ....
    +- 162[this is a=>これは、] +- 17508[pen .=>ペン である。]
    +- ...
    ...
```

```
creating hypothesis 1 from 0
    base score 0
    translation cost -1.35766
    distortion cost 0
    language model cost for 'ここ' -0.492322
    word penalty 0.361996
    score -1.48798 + futureCost -1.92873 = -3.41672
```

new best estimate for this stack

merged hypothesis on stack 1, now size 1

```
creating hypothesis 2 from 0
    base score 0
    translation cost -1.04392
    distortion cost 0
    language model cost for 'その' -0.522497
    word penalty 0.361996
    score -1.20443 + futureCost -1.92873 = -3.13316
```

new best estimate for this stack

merged hypothesis on stack 1, now size 2

....

```
creating hypothesis 162 from 0 <=====
    base score 0
    translation cost -1.03133
    distortion cost 0
    language model cost for 'これ' -0.437817
    language model cost for 'は' -0.219777
    language model cost for ', ' -0.0623787
    word penalty 1.08599
    score -0.665315 + futureCost -1.08455 = -1.74987
```

new best estimate for this stack

better path, overwriting existing hypothesis 156

```
creating hypothesis 163 from 0
    base score 0
    translation cost -1.85256
    distortion cost 0
```

language model cost for 'これ' -0.437817
language model cost for 'は' -0.219777
language model cost for '、' -0.0623787
language model cost for 'いわゆる' -0.876915
word penalty 1.44798
score -2.00147 + futureCost -1.08455 = -3.08602
merged hypothesis on stack 3, now size 8

.....

creating hypothesis 174 from 1
base score -1.48798
translation cost -2.11081
distortion cost 0
language model cost for 'さ' -1.18326
language model cost for 'れ' -0.155323
language model cost for 'て' -0.0927082
word penalty 1.08599
score -3.9441 + futureCost -1.93378 = -5.87788

merged hypothesis on stack 2, now size 45

creating hypothesis 175 from 1
base score -1.48798
translation cost -1.48134
distortion cost 0
language model cost for 'に' -0.46083
word penalty 0.361996
score -3.06816 + futureCost -1.93378 = -5.00194

merged hypothesis on stack 2, now size 46

creating hypothesis 176 from 1
base score -1.48798
translation cost -1.47625
distortion cost 0
language model cost for 'も' -1.18326
word penalty 0.361996
score -3.7855 + futureCost -1.93378 = -5.71929

merged hypothesis on stack 2, now size 47

creating hypothesis 177 from 1
base score -1.48798
translation cost -1.29528
distortion cost 0
language model cost for 'れる' -1.18326
word penalty 0.361996
score -3.60453 + futureCost -1.93378 = -5.53832

merged hypothesis on stack 2, now size 48

.....

```

creating hypothesis 17508 from 162 <=====
    base score -0.665315
    translation cost -0.68607
    distortion cost 0
    language model cost for 'ペン' -1.06335
    language model cost for 'で' -0.412911
    language model cost for 'ある' -0.29139
    language model cost for '。' -0.0360304
    adding word </s> -5.05187e-07
    word penalty 1.44798
    score -1.70708 + futureCost 0 = -1.70708
new best estimate for this stack
better path, overwriting existing hypothesis 17467
creating hypothesis 17509 from 157
    base score -1.33158
    translation cost -1.09842
    distortion cost 0
    language model cost for '筆' -1.18326
    word penalty 0.361996
    score -3.25127 + futureCost -0.713159 = -3.96443
better path, overwriting existing hypothesis 10164
creating hypothesis 17510 from 157
    base score -1.33158
    translation cost -1.00945
    distortion cost 0
    language model cost for 'ペン' -1.18326
    language model cost for 'の' -0.426796
    word penalty 0.723992
    score -3.2271 + futureCost -0.713159 = -3.94026
worse than existing path, discarding

....

best hypothesis now 0
best hypothesis now 2
best hypothesis now 18
[ 17508 => 162 ]
[ 162 => 0 ]
BEST: これは、 |0.514112|0|2| ペン である。 |0.35283|3|4| -1.70708

```

まとめ

- フレーズベースのSMTのデコーダの基本動作は簡単である
- デコーダの研究は，これからである．