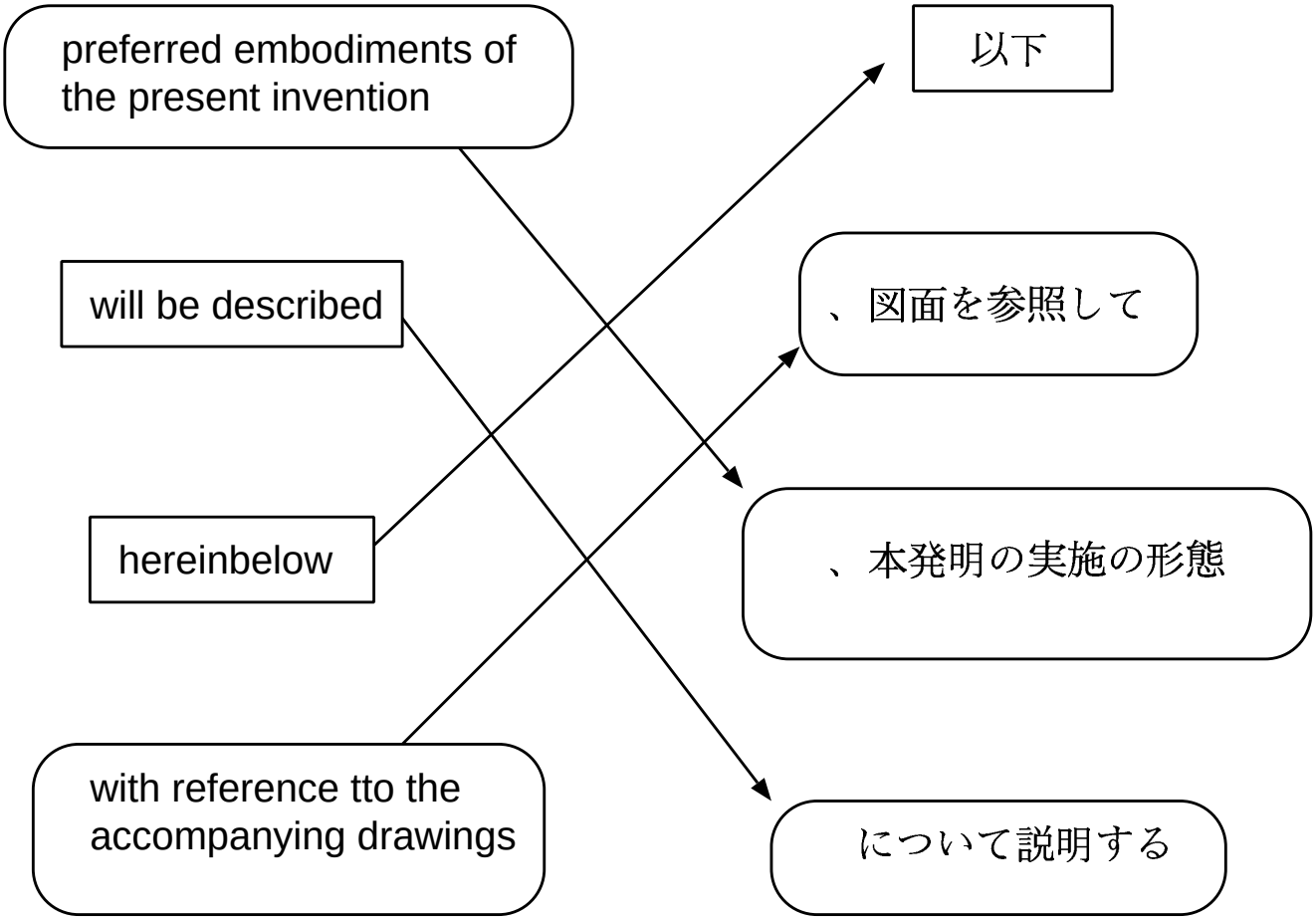
17. 句に基づく統計的機械翻訳(SMT)における素性

内山将夫@NICT
mutiyama@nict.go.jp

句に基づく統計的機械翻訳 (SMT) における素性

- 句単位の翻訳の例
- 句単位の翻訳を構成する基本要素
- 句単位の翻訳の素性

句単位の英日翻訳の例



- 英語の文を句にわける
  (e.g., will be described)
- 各句を翻訳する
  (e.g., について説明する)
- 句を並べ替える

句単位の翻訳の基本要素

- 英語の文$\mathbf{e}$を句にわける

$$\mathbf{e} = \overline{e}_1 \overline{e}_2 \ldots \overline{e}_l$$

- 各句$\overline{e}_i$を日本語の句$\overline{n}_i$に翻訳する

$$\mathbf{n} = \overline{n}_1 \overline{n}_2 \ldots \overline{n}_l$$

- $\overline{n}_i$を並べかえる.

句単位の翻訳のための素性

原言語を $\mathbf{f}$，対象言語を $\mathbf{e}$ としたとき，

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

なる $\hat{\mathbf{e}}$ を探したい．そのために，$\mathbf{e}$ と $\mathbf{f}$ の組をスコア付けするために，素性 $h_i(\mathbf{e}, \mathbf{f})$ を利用する．

素性の例

- 言語モデル
- 句単位の翻訳モデル
- 句を構成する単語に基づく翻訳モデル
- 単語ペナルティ
- 句ペナルティ
- 語順ペナルティ

これらの素性の値は，$\mathbf{e}, \mathbf{f}$ より $h_i(\mathbf{e}, \mathbf{f})$ として計算される．各素性の重みは，自動推定する (後述).

言語モデル

$$
\begin{aligned}
h(\mathbf{e}, \mathbf{f}) &= \log P(\mathbf{e}) \\
&= \log P(e_1, e_2, \ldots, e_l) \\
&= \log \prod_i P(e_i | e_{i-2}, e_{i-1})
\end{aligned}
$$

3-gram 言語モデルにより対象言語 $\mathbf{e}$ の生成確率の対数を素性とする．これが大きい翻訳文 $\mathbf{e}$ は，対象言語としてのつくりが良いと考えられる．

ところで，

3-gram 言語モデルは，前の 2 単語しか見ていないので，とても貧弱なモデルに見えるが，これを越えるモデルはあまりない．良い言語モデルを作ることができれば，機械翻訳に与えるインパクトは大きい．

単語ペナルティと句ペナルティ

単語ペナルティ

$$h(\mathbf{e}, \mathbf{f}) = \mathbf{e}\text{中の単語数}$$

句ペナルティ

$$h(\mathbf{e}, \mathbf{f}) = \mathbf{e}\text{中の句数}$$

これらの重みが正のときには，単語や句がたくさんある$\mathbf{e}$が優先される．負のときには，単語や句は少ない方が良い．
単語ペナルティや句ペナルティは，適切な長さの訳文を出力するために有用である．

句単位の翻訳モデル

$$\begin{aligned} h(\mathbf{e},\mathbf{f}) &= \log P(\mathbf{e}|\mathbf{f}) \\ &= \log \Pi P(\overline{e}|\overline{f}) \\ &= \Sigma \log P(\overline{e}|\overline{f}) \end{aligned}$$

これは $\mathbf{f} \rightarrow \mathbf{e}$ の翻訳モデルだが，逆方向も同様に，

$$h(\mathbf{e},\mathbf{f}) = \Sigma \log P(\overline{f}|\overline{e})$$

ただし，

$$\begin{aligned} P(\overline{f}|\overline{e}) &= \frac{\mathrm{count}(\overline{f},\overline{e})}{\Sigma_{\overline{f}'}\,\mathrm{count}(\overline{f}',\overline{e})} \\ \mathrm{count}(\overline{f},\overline{e}) &= \text{句}\overline{f}\text{と句}\overline{e}\text{が対応している回数} \end{aligned} \quad (1)$$

$P(\overline{f}|\overline{e})$ の問題点

$\mathrm{count}(\overline{f},\overline{e})$ が小さいときに値が信頼できない

$\rightarrow$ スムージング

単語に基づく句対応の重み (双方向)

$$h(\mathbf{e}, \mathbf{f}) = \sum \log \mathrm{lex}(\overline{f}|\overline{e}) \tag{2}$$

$$\mathrm{lex}(\overline{f}|\overline{e}) = \max_{\mathbf{a}} P_w(\overline{f}|\overline{e}, \mathbf{a})$$

$$P_w(\overline{f}|\overline{e}, \mathbf{a}) = \prod_{i=1}^{n} E_w(f_i|\overline{e}, \mathbf{a})$$

$$E_w(f_i|\overline{e}, \mathbf{a}) = \frac{1}{\{j|(i,j) \in \mathbf{a}\}} \sum_{(i,j)\in\mathbf{a}} w(f_i|e_j)$$

$$w(f_i|e_j) = \text{単語対応の確率} = \frac{\mathrm{count}(f_i, e_j)}{\Sigma_{f'} \mathrm{count}(f', e_j)}$$

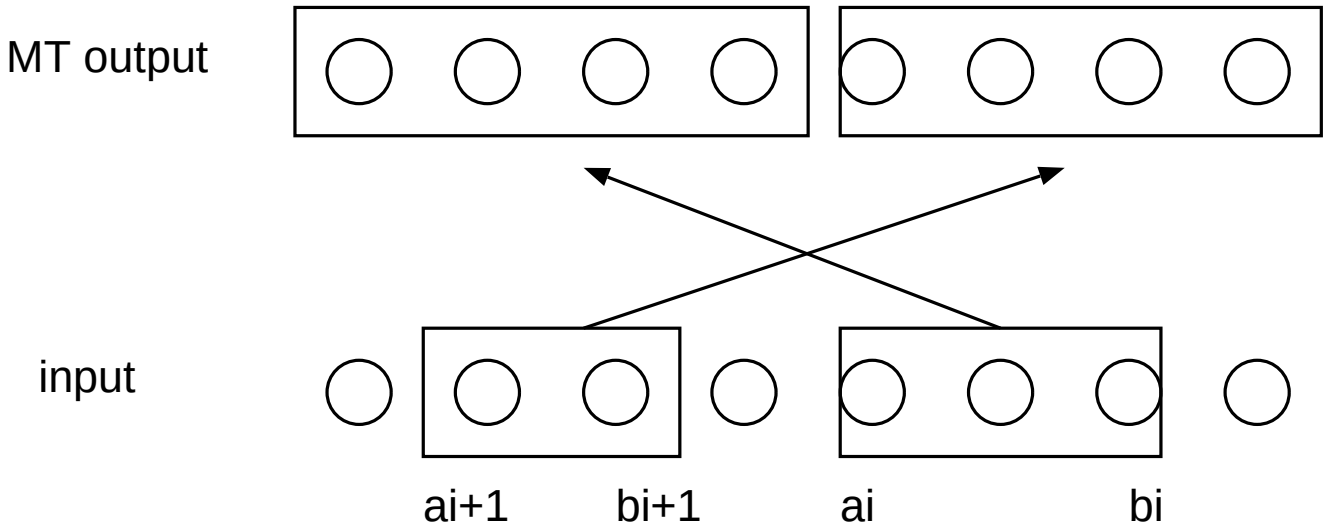
$E_w(f_i|\overline{e}, \mathbf{a}) = w(f_i|e_j)$の$e_j$に関する平均値

$P_w(\overline{f}|\overline{e}, \mathbf{a}) = E_w(f_i|\overline{e}, \mathbf{a})$の積. f_iの条件付き独立を仮定

$\mathrm{lex}(\overline{f}|\overline{e}) = P_w(\overline{f}|\overline{e}, \mathbf{a})$が最大の$\mathbf{a}$に関する確率を採用

語順

語順については，今，盛んに研究されている．単純なものとしては，原言語の語順と異なるものにペナルティを与えるというものがある．



のとき，

$$|b_i - a_{i+1} + 1| \;=\; |\overline{f}_i\text{に対応する}\overline{e}_i\text{の最後の単語位置} \\ -\; \overline{f}_{i+1}\text{に対応す}\overline{e}_{i+1}\text{の最初の単語位置} + 1|$$

として，連続するフレーズが連続するときのペナルティを 0 とし，そうでないときに，離れ具合に応じてペナルティをかける．

まとめ

- 対数線形モデルを利用することにより，様々な素性をスコア付けに利用できる
- 各素性は，それぞれ別個に改良できる

特に，

- 言語モデルの改良
- 翻訳モデルの改良
  - カバー率の高い句表をつくる
  - 良いスコア付けをする
- 語順の制約

が重要である．