MASTER RESEARCH INTERNSHIP IN COMPUTER SCIENCE

Machine learning, Information and Content

"Unsupervised Neural Word Alignment HMM"

Computer Science Laboratory for Mechanics and Engineering Sciences (LIMSI)

Author:
VU Trong-Bach

Supervisor:
Dr. François YVON

Contents

I. ISSUES INTRODUCTION

I. ISSUES INTRODUCTION

Machine Translation

IBM 1

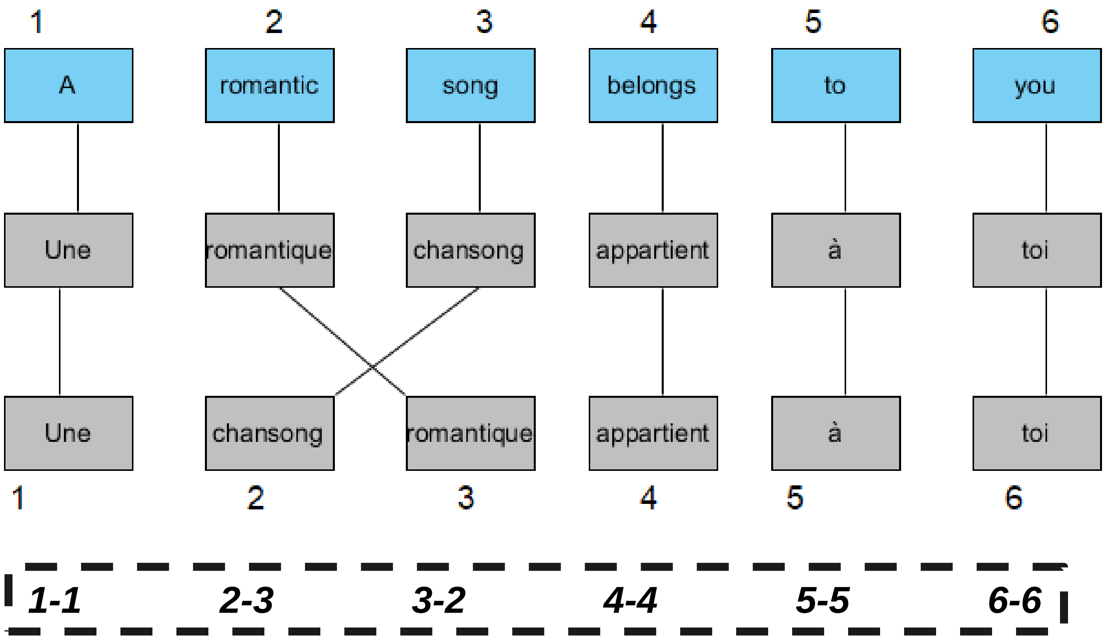
IBM 2

IBM 3

IBM 4

IBM 5

IBM 6



Word Alignment*

* [Och and Ney, 2003] [Vogel et al., 1996]

I. ISSUES INTRODUCTION

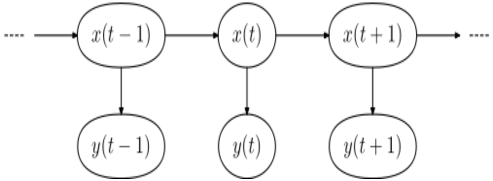
Machine Translation

$$f^J = \{f_1 \cdots f_j \cdots f_J\}$$
$$e^I = \{e_1 \cdots e_i \cdots e_I\}$$

$$e^I = arg\max_{e^I}\{P(e^I|f^J)\}$$
$$= arg\max_{e^I}\{P(e^I) \cdot P(f^J|e^I)\}$$

$$P(f^J|e^I) = \sum_{a^J} P(f^J, a^J|e^I)$$
$$= \sum_{a^J} \prod_{j=1}^{J} P(f_j, a_j|f_{j-1}, a_{j-1}, e^I)$$
$$= \sum_{a^J} \prod_{j=1}^{J} P(a_j|f_{j-1}, a_{j-1}, e^I) \cdot P(f_j|f_{j-1}, a_j, e^I)$$

<u>*First order dependence*</u>



Hidden Markov Models

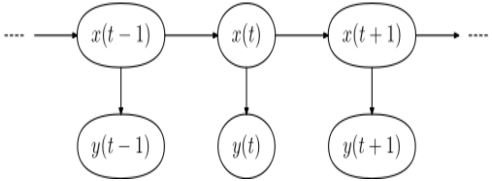
$$\boxed{P(f^J|e^J) = \sum_{a^J} \prod_{j=1}^{J} [p(a_j|a_{j-1}, I) \cdot p(f_j|e_{a_j})]}$$

$$P(a_j|f_{j-1}, a_{j-1}, e^I) = p(a_j|a_{j-1}, I)$$
$$P(f_j|f_{j-1}, a_j, e^I) = p(f_j|e_{a_j})$$

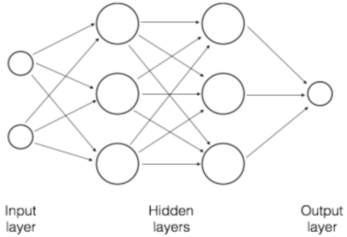
* [Och and Ney, 2003] [Vogel et al., 1996]

I. ISSUES INTRODUCTION

Machine Translation



Hidden Markov Models



Neural Network

$$P(f^J|e^J) = \sum_{a^J} \prod_{j=1}^{J} [\underline{p(a_j|a_{j-1}, I)} \cdot \underline{p(f_j|e_{a_j})}]$$

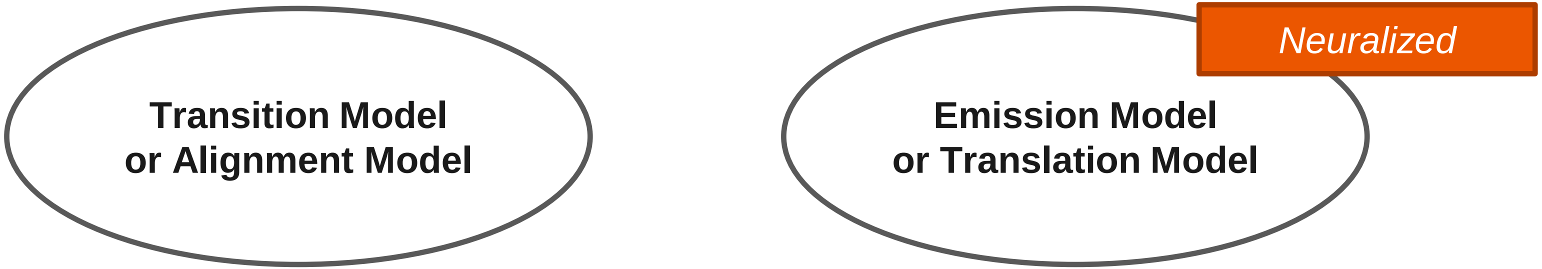
Transition Model or Alignment Model

Neuralized

Emission Model or Translation Model

* [Och and Ney, 2003] [Vogel et al., 1996]

II. PROPOSED METHOD



II. PROPOSED METHOD

Transition Model

$$p(a_j|a_{j-1}, I) \text{ or } p(i|i', I) = \frac{s(i-i')}{\sum_{i''=1}^{I} s(i''-i')}$$

Using a non-negative set

i'/i	**0**	**1**	**2**	**...**
0	s(0)	s(1)	s(2)	...
1	s(-1)	s(0)	s(1)	s(2)
2	s(-2)	s(-1)	s(0)	s(1)
...	...	s(-2)	s(-1)	s(0)

Empty word problem???

II. PROPOSED METHOD

Transition Model

$$p(i+I|i',I) = p_0 \cdot \delta(i,i')$$
$$p(i+I|i'+I,I) = p_0 \cdot \delta(i,i')$$
$$p(i|i'+I,I) = p(i|i',I)$$

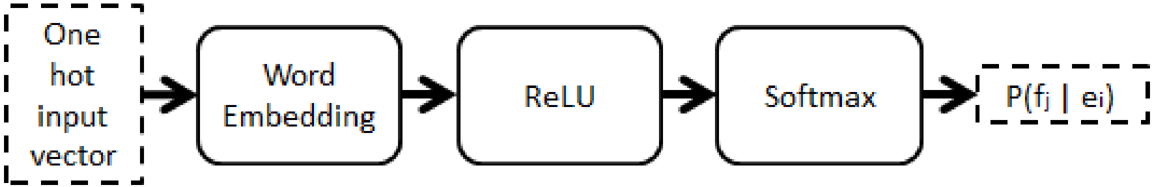
- *p0 is the probability of a transition to the empty word*
- $\delta(i,i')\begin{cases} 1\ if\ i = i' \\ 0\ otherwise \end{cases}$

by extending the HMM empty words e^2I

i'/i	0	1	2	...	0 (I)	1 (I+1)	2 (I+2)	...
0	s(0)	s(1)	s(2)	...	$p_0 \cdot 1$	$p_0 \cdot 0$	$p_0 \cdot 0$	...
1	s(-1)	s(0)	s(1)	s(2)	$p_0 \cdot 0$	$p_0 \cdot 1$	$p_0 \cdot 0$	$p_0 \cdot 0$
2	s(-2)	s(-1)	s(0)	s(1)	$p_0 \cdot 0$	$p_0 \cdot 0$	$p_0 \cdot 1$	$p_0 \cdot 0$
...	...	s(-2)	s(-1)	s(0)	...	$p_0 \cdot 0$	$p_0 \cdot 0$	$p_0 \cdot 1$
0 (I)	s(0)	s(1)	s(2)	...	$p_0 \cdot 1$	$p_0 \cdot 0$	$p_0 \cdot 0$	...
1 (I+1)	s(-1)	s(0)	s(1)	s(2)	$p_0 \cdot 0$	$p_0 \cdot 1$	$p_0 \cdot 0$	$p_0 \cdot 0$
2 (I+2)	s(-2)	s(-1)	s(0)	s(1)	$p_0 \cdot 0$	$p_0 \cdot 0$	$p_0 \cdot 1$	$p_0 \cdot 0$
...	...	s(-2)	s(-1)	s(0)	...	$p_0 \cdot 0$	$p_0 \cdot 0$	$p_0 \cdot 1$

II. PROPOSED METHOD

Emission Model



Unsupervised - How to update θ ???

II. PROPOSED METHOD

Update $\boldsymbol{\theta}$

Maximize the evidence
$p(f|\theta) = \sum_e p(f, e|\theta)$

- To estimate $\boldsymbol{\theta}$, we can use auxiliary function of EM algorithm

$$p(f|\theta) = E_{q(e)}[\ln p(f, e|\theta)] + H[q(e)] + KL(q(e)||p(f, e|\theta))$$

We choose:

- $q(e)$ *to be posterior* $p(e|f)$
- $H[q(e)]$ *a constant -> dropped*
- Setting KL *divergence to zero*

Only maximize $E_{p(e|f)}[\ln p(f, e|\theta)]$

Gradient:

$$J(\theta) = \sum_e p(e|f) \frac{\partial}{\partial\theta} \ln p(f, e|\theta)$$

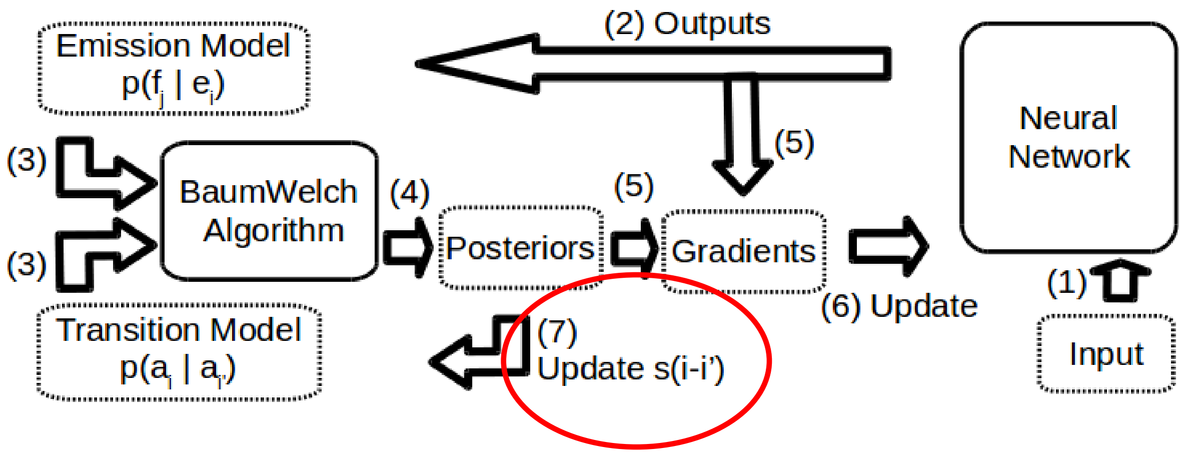
$$J(\theta) = \sum_j \sum_{a_j} p(a_j|f_j) \frac{\partial}{\partial\theta} \ln p(f_j|e_{a_j}, \theta)$$

How calculate posteriors ???

II. PROPOSED METHOD

$$p(a_j = i' | f_j, \theta) \propto \alpha_{i'}(j)\beta_{i'}(j)$$

$$p(a_j = i', a_{j+1} = i | f^J, \theta) \propto \alpha_{i'}(j) p(a_{j+1} = i | a_j = i') \times \beta_i(j+1) p(f_{j+1} | e_{a_j})$$



Global Model

II. PROPOSED METHOD

Update non-negative set s(i-i')

$$p(a_j = i' | f_j, \theta) \propto \alpha_{i'}(j) \beta_{i'}(j)$$

$$p(a_j = i', a_{j+1} = i | f^J, \theta) \propto \alpha_{i'}(j) p(a_{j+1} = i | a_j = i') \times \beta_i(j+1) p(f_{j+1} | e_{a_j})$$

$$s(i, i') = \frac{\sum_{n}^{N} \sum_{j}^{J-1} p_n(a_j = i', a_{j+1} = i | f^J, \theta)}{\sum_{n}^{N} \sum_{j}^{J-1} p_n(a_j = i' | f_j, \theta)}$$

N is the number of pair sentences (f; e) in the corpus

III. EXPERIMENTS

Evaluation Methodology

Viterbi Alignment

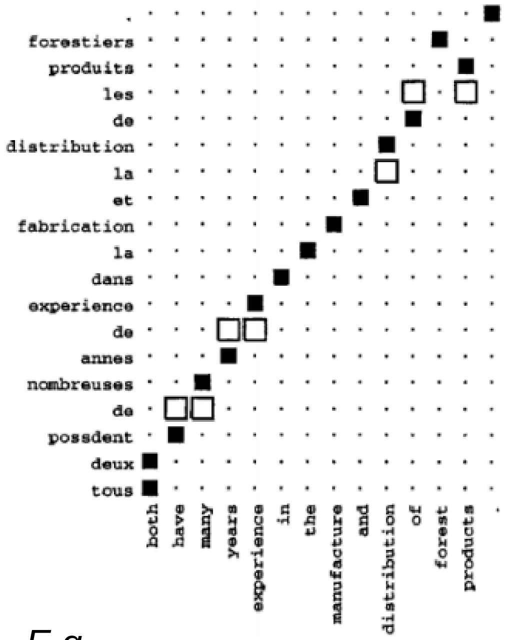
$$\hat{a}^J = \arg\max_{a^J} p(f^J, a^J | e^{2I})$$

$$= \arg\max_{a^J} \left\{ \prod_{j=1}^{J} [p(a_j | a_{j-1}, 2I) \cdot p(f_j | e_{a_j}^{2I})] \right\}$$

$$= \left[\arg\max_{a_j} \{ p(a_j | a_{j-1}, 2I) \cdot p(f_j | e_{a_j}^{2I}) \} \right]_{j=1}^{J}$$

$$\boxed{AER = 1 - \frac{(|A \cap S| + |A \cap P|)}{(|A| + |S|)}}$$

The best AER = 0.0,
where S (sure alignments), P (possible alignments)
A (hypothesis alignments).



E.g.
Sure: 1-1 2-1 3-2 4-3 5-4 …
Possible: 4-2 4-3 7-4 7-5 ….

[Och and Ney, 2003]

III. EXPERIMENTS

Corpus	Type	Name	No Sentences
Roman-English	Training	Naacl2003	48k
	Training	WMT2016 SETIMES	213k
	Testing	Naacl2003	248
English-Czech	Training	News commentary v.11	191432
	Testing	Marecek2008	2500
Dutch-English	Training	Europarl	2M
	Testing	Europarl	509
English-Italian	Training	Europarl	2M
	Testing	WAGS	6700

- Romanian-English testing set only includes sure alignments.
- English-Italian testing set includes only rare words.

Corpus	IBM2	IBM4	Best
Ro-En	30.7	30.4	IBM4
En-Cz	24.3	26.7	IBM2
Du-En	27.4	22.3	IBM4
En-It	68.6	80.6	IBM2

IBM2: Fast_align
IBM4: MGIZA++

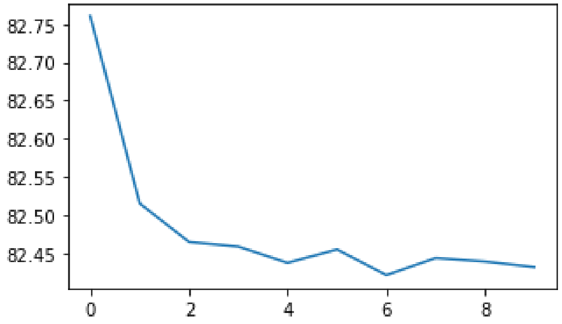
III. EXPERIMENTS

Corpus	IBM2	IBM4	NWA-HMM
En-Cz	24.3	26.7	82.4

- ○ 82.4 is still a very bad score.
- ✓ AER score has a decreasing tendency until 9^{th} epoch

Other investigations:

- ✓ Maximum log-likelihood in Baum-Welch algorithm.
- ✓ The variation of non-negative transition elements s(i-i‘) through epochs.

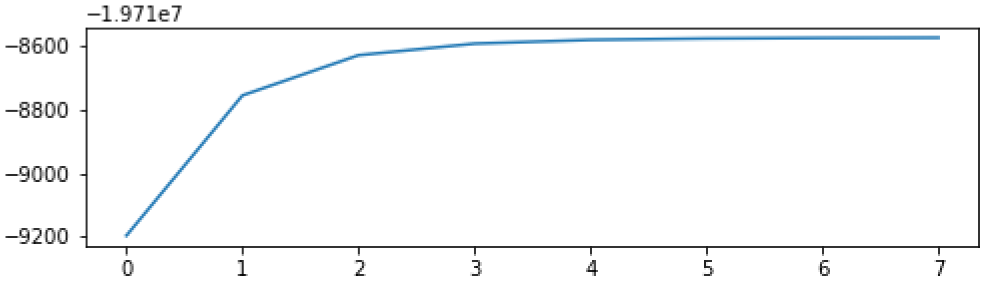


The AER scores for 10 first epochs on English-Czech corpus

Maximum log-likelihood

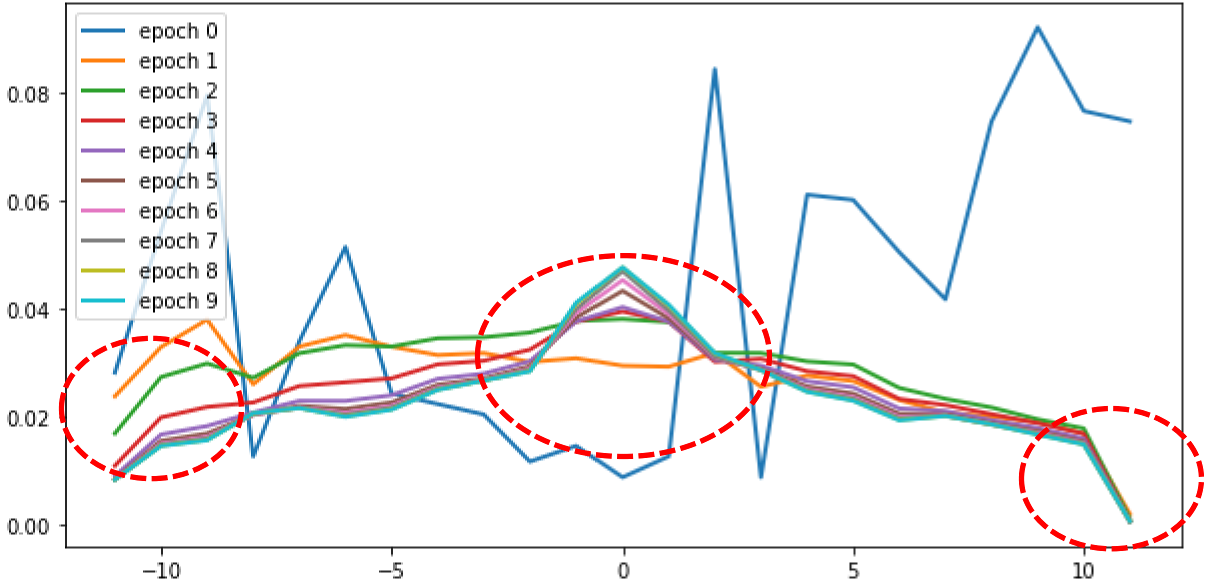
BW Algorithm uses EM algorithm to find the maximum log-likelihood:

$$\theta^* = arg\max_{\theta} P(f^J|\theta)$$



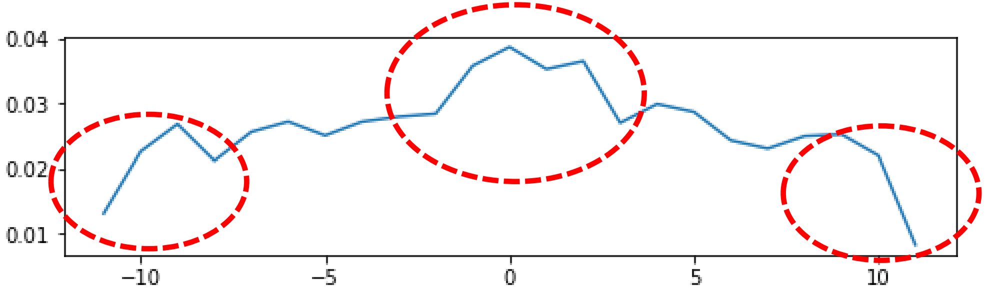
✓ Ascending trend through epochs

The variation of non-negative transition elements s(i-i')



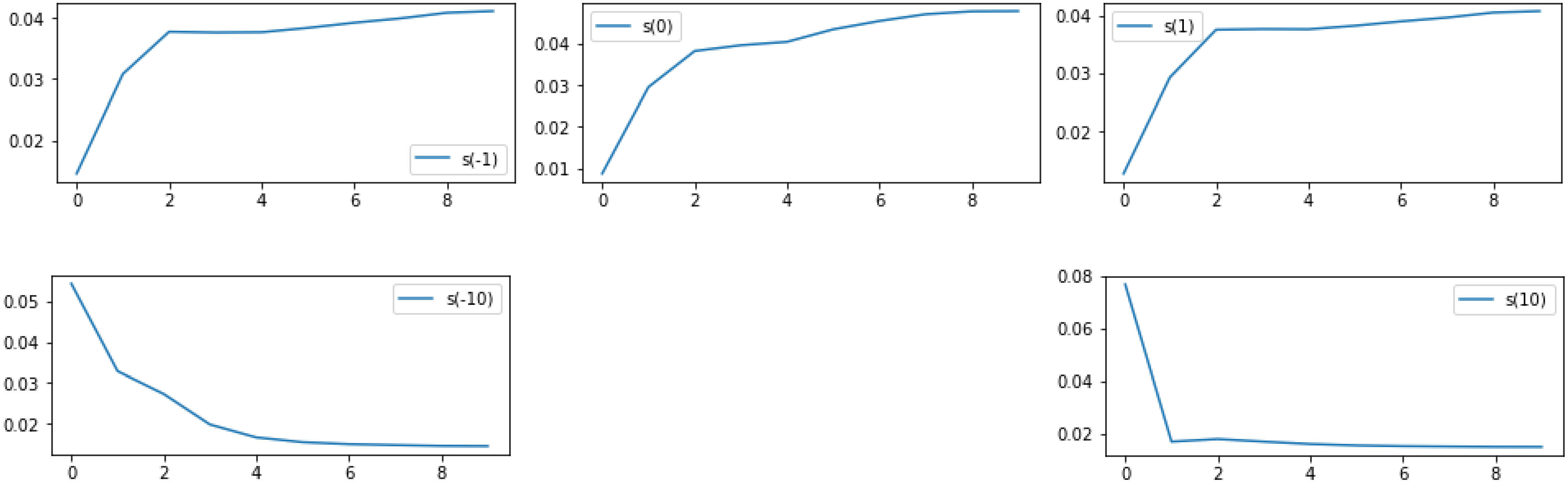
- ✓ The sorter distance the higher value

The mean of non-negative transition elements s(i-i')

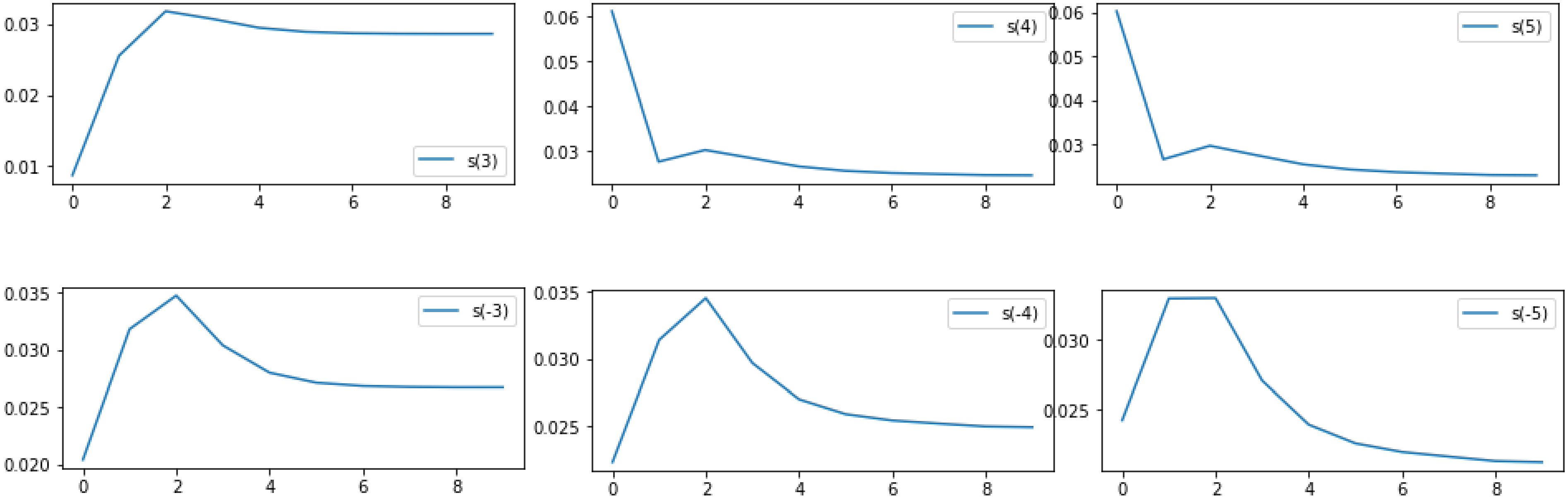


- ✓ The sorter distance the higher value

The variation of non-negative transition elements s(i-i')



The variation of non-negative transition elements s(i-i')



Programing Tips and Tricks

A potential arithmetical issue during running BW-Alg
e.g. $x^{-200}.x^{-200}$

1. **Baum-Welch normalization**

2. **Log-space multiplication between two very small numbers dealing with normalization task**

Baum-Welch normalization

Forward messages α and backward messages β can get very small. e.g. x^-200.

Solution: Using the same normalization factor

$$Z(j) = \sum_{i}^{I} \alpha_i(j)$$

$$\hat{\alpha}_i(j) = \alpha_i(j)/Z(j)$$

$$\hat{\beta}_i(j) = \beta_i(j)/Z(j)$$

Log-space multiplication between two very small numbers dealing with normalization task

$$\log(\hat{x}_i) = \log(x_i) - \log(\textstyle\sum_{j=0}^{J}(x_j)).$$

However $\displaystyle\sum_{j=0}^{J}(\log(x_j)) \neq \log(\sum_{j=0}^{J}(x_j))$

$$\hat{x}_i = \frac{x_i}{\sum_{j=0}^{J} x_j}$$

$$\log(\sum_{j=0}^{J}(x_j)) = \log(x_0) + \log(1 + \sum_{j=1}^{J}\frac{x_j}{x_0})$$

$$= \log(x_0) + \log(1 + \sum_{j=1}^{J}(\exp^{\log(x_j) - log(x_0)}))$$

$$x_i = a_i \times b_i$$

$$= \log(a_0 \times b_0) + \log(1 + \sum_{j=1}^{J}(\exp^{\log(a_j \times b_j) - log(a_0 \times b_0)}))$$

$$\log(x_i) = \log(a_i) + \log(b_i)$$

$$= \log(a_0) + \log(b_0) + \log(1 + \sum_{j=1}^{J}(\exp^{\log(a_j) + \log(b_j) - log(a_0) - \log(b_0)}))$$

where $x_0 > x_1 > ... > x_J$ are sorted in descending order.

IV. CONCLUSION

For enhancing translation performance:

- ✓ Believe that an improvement of this proposed method could be useful for further work in unsupervised neural alignment.
- ✓ Potentially to be integrated into well-know Attention Model.

Reasons for using neural alignment:

- ✓ Do not have much aligned corpus which requires expensively human resources.
- ✓ Have not yet found an unsupervised efficient automatic word alignment method which could obtain less than about 10% of error.
- ✓ The word alignment is still a promised model to enhance the translation achievements considerably.
- ✓ Neural network itself is really powerful for extracting special features on our data that plays a necessary role in each alignment

Thank you for your attention !