Sampling-based Alignment and Hierarchical Sub-sentential Alignment in Chinese–Japanese Translation of Patents

Wei Yang, Zhongwen Zhao, Baosong Yang and Yves Lepage

Graduate School of Information, Production and Systems
Waseda University

{kevinyoogi@akane., zzw890827@fuji., yangbaosong@fuji.}waseda.jp, yves.lepage@waseda.jp

This paper describes Chinese–Japanese translation systems based on different alignment methods using the JPO corpus and our submission (ID: WASUIPS) to the subtask of the 2015 Workshop on Asian Translation. One of the alignment methods used is bilingual hierarchical sub-sentential alignment combined with sampling-based multilingual alignment. We also accelerated this method and in this paper, we evaluate the translation results and time spent on several machine translation tasks. The training time is much faster than the standard baseline pipeline (GIZA++/Moses) and MGIZA/Moses.

Bilingual hierarchical sub-sentential alignment method used in Phrase-based Statistical Machine Translation (PB-SMT)

- Associative approaches: use a local maximization process in which each sentence is processed independently.
  - Anymalign[1]: is an open source multilingual associative aligner (Lardilleux and Lepage, 2009; Lardilleux et al., 2013). This method samples large numbers of sub-corpora randomly to obtain source and target word or phrase occurrence distributions.
  - Cutnalign: is a bilingual hierarchical sub-sentential alignment method (Lardilleux et al., 2012). It is based on a recursive binary segmentation process of the alignment matrix between a source sentence and its corresponding target sentence. We make use of this method in combination with Anymalign. It is a three-step approach:
    - measure the strength of the translation link between any source and target pair of words;
    - compute the optimal joint clustering of a bipartite graph to search the best alignment;
    - segment and align a pair of sentences.

When building alignment matrices, the strength between two words is evaluated using the following formula (Lardilleux et al., 2012).

$$w(s,t) = p(s|t) \times p(t|s) \tag{1}$$

($p(s|t)$ and $p(t|s)$) are translation probabilities estimated by Anymalign. An example of alignment matrix is shown in Table 1.

	それら	の	値	に	基づい	て	upgma	法	によって	クラ	スター	分析	を	行っ	た	。
根据	ε	ε	ε	0.27	0.46	0.01	ε	ε	0.002	ε	ε	ε	ε	ε	ε	0.02
这些	0.38	ε	ε	0.02	ε	ε	ε	0.001	ε	ε	ε	ε	ε	ε	ε	0.01
值	0.012	0.27	0.44	ε	ε	ε	ε	ε	ε	ε	ε	ε	ε	ε	ε	0.03
，	0.002	0.01	0.01	0.13	0.12	0.21	0.10	0.002	0.001	0.002	0.001	0.01	0.01	0.01	ε	0.01
通过	ε	ε	0.01	ε	ε	0.06	ε	ε	0.52	ε	ε	ε	ε	0.02	ε	0.01
upgma	ε	ε	ε	ε	ε	ε	0.75	ε	ε	ε	ε	ε	ε	ε	ε	0.02
法	ε	ε	ε	ε	ε	ε	ε	0.013	0.013	ε	ε	ε	ε	ε	ε	0.01
进行	ε	ε	ε	ε	ε	ε	ε	ε	ε	ε	ε	0.01	0.23	0.34	0.21	0.01
聚类	ε	ε	ε	ε	ε	ε	ε	ε	ε	0.045	0.045	ε	ε	ε	ε	0.02
分析	ε	ε	ε	ε	ε	ε	ε	ε	ε	ε	ε	0.5	ε	ε	ε	0.01
。	0.01	0.02	0.01	0.02	0.01	0.01	0.02	0.01	0.02	0.01	0.002	0.01	0.02	0.01	0.01	0.7

Table 1: *An example of an alignment matrix which contains the translation strength for each word pair (Chinese–Japanese). The scores are obtained using Anymalign's output. Computing by w.*

The optimal joint clustering of a bipartite graph is computed recursively using the following formula for searching the best alignment between words in the source and target languages (Zha et al., 2001; Lardilleux et al., 2012).

$$cut(X,Y) = W(X,\overline{Y}) + W(\overline{X},Y) \tag{2}$$

X, $\overline{X}$, Y, $\overline{Y}$ denote the segmentation of the sentences. Here the block we start with is the entire matrix. Splitting horizontally and vertically into two parts gives four sub-blocks.

$$W(X,Y) = \sum_{s \in X, t \in Y} w(s,t) \tag{3}$$

$W(X,Y)$ is the sum of all translation strengths between all source and target words inside a sub-block (X,Y).
The point where to is found on the x and y which minimize $Ncut$ (Lardilleux et al., 2012):

$$Ncut(X,Y) = \frac{cut(X,Y)}{cut(X,Y) + 2 \times W(X,Y)} + \frac{cut(\overline{X},\overline{Y})}{cut(\overline{X},\overline{Y}) + 2 \times W(\overline{X},\overline{Y})} \tag{4}$$

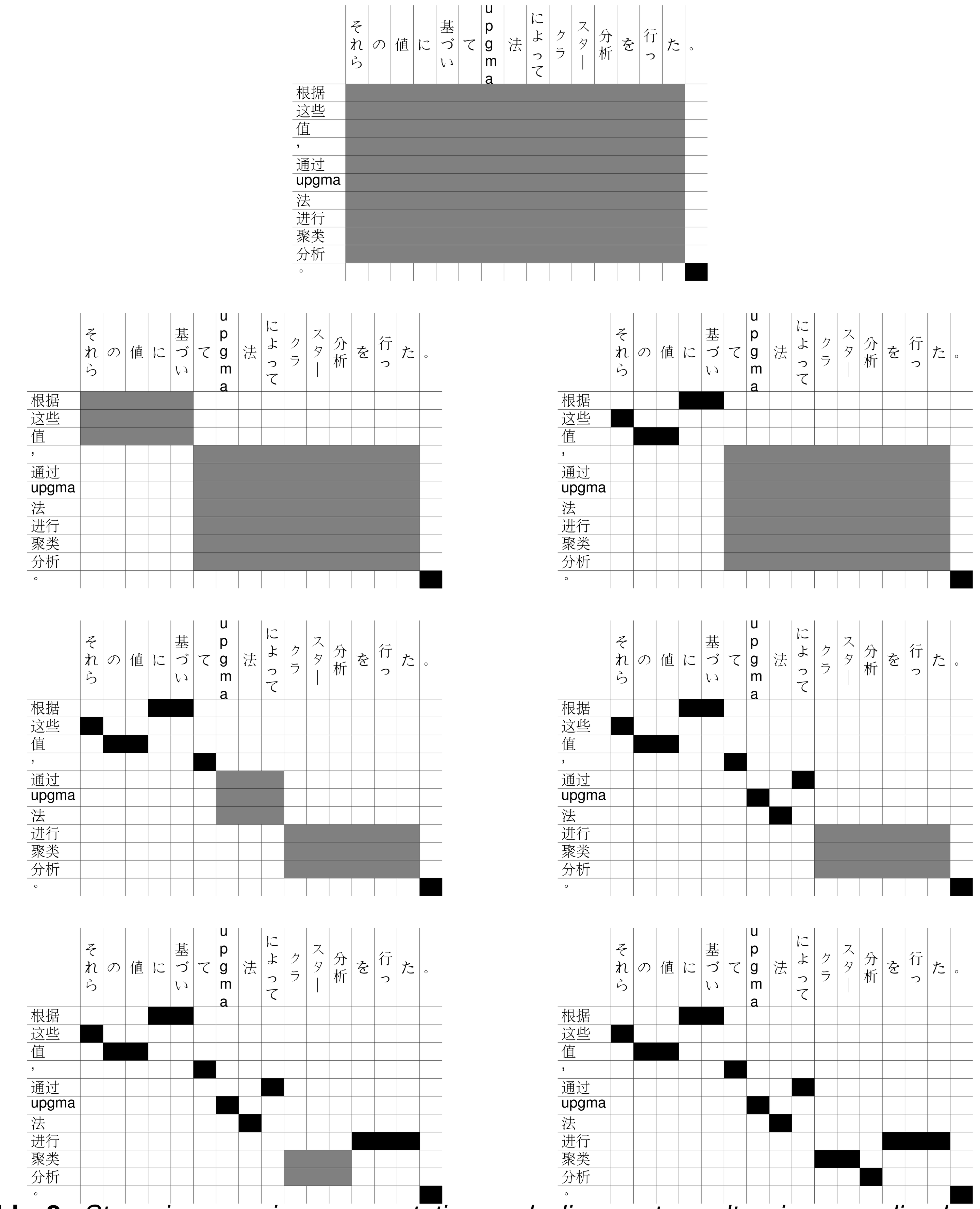

Table 2: *Steps in recursive segmentation and alignment result using sampling-based alignment and hierarchical sub-sentential alignment method.*

SMT experiments

- **Experimental protocol (Chinese and Japanese data used)**: Chinese–Japanese JPO Patent Corpus (JPC)[2] provided by WAT 2015 for the patents subtask. We used sentences of 40 words or less than 40 words as our training data for the translation models, but use all of the Japanese sentences in the parallel corpus for training the language models. We used all of the development data for tuning.

	Baseline	Chinese	Japanese
train	sentences	820,184	820,184
	words	15,655,674	20,279,246
	mean $\pm$ std.dev.	19.39 $\pm$ 6.71	25.08 $\pm$ 7.75
tune	sentences	4,000	4,000
	words	114,363	143,853
	mean $\pm$ std.dev.	28.71 $\pm$ 18.34	36.12 $\pm$ 21.73
test	sentences	2,000	2,000
	words	55,582	70,117
	mean $\pm$ std.dev.	27.83 $\pm$16.73	35.09 $\pm$ 20.16

- **Experimental results**
  (using the different alignment approaches, tools and Moses versions)
  - alignment tools: GIZA++ (baseline) and MGIZA, moses 2.1.1.

s→t	Moses	Aligner	BLEU	RIBES	Training time
zh→ja	2.1.1	MGIZA	37.70	0.783000	5:34:28
	2.1.1	GIZA++	37.46	0.778914	4:43:56

  - alignment tools: the alignment method of combining sampling-based alignment and bilingual hierarchical sub-sentential alignment methods. Here, 2 (c) shows option -i of Anymalign is 2, and Cutnlaign version where core component is implemented in C.

Language	Moses	Aligner		BLEU	Training time
		Anymalign + Cutnalign			
		Timeout (s)	i		
zh-ja	3.0	1200	2 (c)	36.11	1:2:8
zh-ja	3.0	5400	2 (c)	36.07	2:9:29
zh-ja	2.1.1	1200	2 (c)	35.95	0:57:1
zh-ja	2.1.1	1200	2 (python)	35.93	1:1:16

Conclusion

We have shown that it is possible to accelerate development of SMT systems following the work by Lardilleux et al. (2012) and Yang and Lepage (2015) on bilingual hierarchical sub-sentential alignment. We performed several machine translation experiments using different alignment methods and obtained a significant reduction of processing training time. Setting different timeouts for Anymalign did not change the translation quality. In other word, we get a relative steady translation quality even when less time is allotted to word-to-word association computation. Here, the fastest training time was only 57 minutes, one fifth compared with the use of GIZA++ or MGIZA.

[1] https://anymalign.limsi.fr
[2] http://lotus.kuee.kyoto-u.ac.jp/WAT/patent/index.html