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

Abstract

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.

1 Introduction

Phrase-based Statistical Machine Translation (PB-SMT) as a data-oriented approach to machine translation has been widely used for over 10 years. The Moses (Koehn et al., 2007) open source statistical machine translation toolkit was developed by the Statistical Machine Translation Group at the University of Edinburgh. During the three processes (training, tuning and decoding) for building a phrase-based translation system using Moses, training is the most important step as it creates the core knowledge used in machine translation. Word or phrase alignment in the training step allows to obtain translation relationships among the words or phrases in a sentence-aligned bi-corpus. Word or phrase alignment affects the quality of translation. It is also one of the most time-consuming processing step.

The *probabilistic* approach attempts at determining the best set of alignment links between source and target words or phrases in parallel sentences. IBM models (Brown et al., 1993) and HMM alignment models (Vogel et al., 1996), which are typical implementation of the EM algorithm (Dempster et al., 1977), are the most widely used representatives in this category. GIZA++ (Och and Ney, 2003) implemented IBM Models, it aligns words based on statistical models. It is a global optimization process simultaneously considers all possible associations in the entire corpus and estimates the parameters of the parallel corpus. Several improvements were made: MGIZA (Gao and Vogel, 2008) is a parallel implementation of IBM models. However, the parallelization may lead to slightly different final alignment results, thus preventing reproduction of results to a certain extent.

The *associative* approaches, introduced in (Gale and Church, 1991), do not rely on an alignment model, but on independence statistical measures. The Dice coefficient, mutual information (Gale and Church, 1991), and likelihood ratio (Dunning, 1993) are representative cases of this approach. The associative approaches use a local maximization process in which each sentence is processed independently. Sampling-based multilingual alignment (*Anymalign*) (Lardilleux et al., 2013) and hierarchical sub-sentential alignment (*Cutnalign*) (Lardilleux et al., 2012) are two associative approaches.

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. The more often two words or phrases have the same occurrence distribution over particular sub-corpora, the higher the association between them.

We can run Anymalign by setting with -t (running time) option and stop it at any time, and the option -i allows to to extract longer phrases

[1] https://anymalign.limsi.fr

Proceedings of the 2nd Workshop on Asian Translation (WAT2015), pages 87–94,
Kyoto, Japan, 16th October 2015.

by enforcing n-grams to be considered as tokens. For pre-segmented texts, option -i allows to group words into phrases more easily.

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.

In the experiments, reported in this paper, we extend the work to decrease time costs in the training step. We obtained comparable results in only one fifth of the training time required by the GIZA++/Moses baseline pipeline.

2 Chinese and Japanese data used

The data used in our systems are the Chinese–Japanese JPO Patent Corpus (JPC)[2] provided by WAT 2015 for the patents subtask (Nakazawa et al., 2015). It contains 1 million Chinese–Japanese parallel sentences in four domains in the training data. These are Chemistry, Electricity, Mechanical engineering, and Physics. 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. For Chinese and Japanese segmentation we used the Stanford Segmenter (version: 2014-01-04 with Chinese Penn Treebank (CTB) model)[3] and Juman (version 7.0)[4]. Table 1 shows some statistics on the data we used in our systems (after tokenization, lowercase and clean).

3 Bilingual hierarchical sub-sentential alignment method

Cutnalign as a bilingual hierarchical sub-sentential alignment method based on a recursive binary segmentation process of the alignment matrix between a source sentence and its translation. 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.

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

Table 1: Statistics of our baseline training data of JPC.

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 2.

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):

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

[2] http://lotus.kuee.kyoto-u.ac.jp/WAT/patent/index.html

[3] http://nlp.stanford.edu/software/segmenter.shtml

[4] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN

Table 3 shows several segmentations out of all the possible segmentation in two blocks by computing the sub-sentential alignment between a Chinese and a Japanese sentences. For each word pair (x, y), we compute $Ncut(x, y)$. In this case, we start at word pair (根据, それら), the search space is the rectangle area [(根据, それら), (。, 。)]. In Table 3, only 7 out of all the possible segmentations in two blocks are shown. The number of possible segmentation is: the length of the Japanese sentence minus one, multiplied by the length of the Chinese sentence minus one, multiplied by two, as there are two possible direction for segmenting. After computing all $Ncut(x, y)$, we compare and find the minimal $Ncut(x, y)$. Table 4 shows the flow of recursive segmentation and alignment.

In the our experiments, we introduced two types of improvements (Yang and Lepage, 2015) compared to the original implementation. The first one, introduces multi-processing in both the sampling-based alignment method and hierarchical sub-sentential alignment method so as to trivially accelerate the overall alignment process. We also re-implement the core of *Cutnalign* in C. The second one, approximations in the computation of $Ncut$ accelerate some decisions. Also a method to reduce the search space in hierarchical sub-sentential alignment has been introduced, so that important speed-ups are obtained. We refer the reader to (Yang and Lepage, 2015) for a detailed description of these improvements.

4 Experiments based on different alignment methods

4.1 Experiment settings

Here, we basically perform experiments with GIZA++ or MGIZA. The phrase tables are extracted from the alignments obtained using the grow-diag-final-and heuristic (Ayan and Dorr, 2006) integrated in the Moses toolkit. Our sampling-based alignment method and hierarchical sub-sentential alignment method are also evaluated within a PB-SMT system built by using the Moses toolkit, the Ken Language Modeling toolkit (Heafield, 2011) and a lexicalized reordering model (Koehn et al., 2005). We built systems from Chinese to Japanese. Each experiment was run using the same data sets (see Section 2). Translations were evaluated using BLEU (Papineni et al., 2002) and RIBES (Isozaki et al., 2010).

We used *Anymalign* (i=2, two words can be considered as one token) and *Cutnalign* to build phrase tables. As a timeout (-t) should be given, we set two different timeouts (5400 sec. and 1200 sec.). We also use different *Cutnalign* versions where core components are implemented in C or Python. We passed word-to-word associations output by Anymalign (i=2) to *Cutnalign* which produces sub-sentential alignments, which are in turn passed to the grow-dial-final-and heuristic of the Moses toolkit to build phrase tables.

4.2 Results

Evaluation results using different alignment methods based on the same data sets are given in Tables 5 and 7. The system built based on GIZA++/Moses pipeline as a baseline system is given in Table 5. We also show the evaluation results obtained by the WAT 2015 automatic evaluation[5] in Table 6 and 8. The results in Table 7 and 8 show that there are no significant differences among the evaluation results based on different versions of Moses, different *Anymalign* timeouts or different versions of *Cutnalign*. However, the training times changed considerably depending on the timeouts for Anymalign. The fastest training time is obtained with Moses version 2.1.1, a timeout of 1200 sec. for *Anymalign* and the C version of *Cutnalign*: 57 minutes, i.e., about one fifth of the time used by GIZA++ or MGIZA (Table 5 and 6). We also checked the confidence intervals between using GIZA++ and our method (the fastest one): 37.24 ± 0.86 and 35.72 ± 0.90. The probability of actually getting them (p-value) is 0.

5 Conclusion

In this paper, 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.

[5] http://orchid.kuee.kyoto-u.ac.jp/WAT/

Here, the fastest training time was only 57 minutes, one fifth compared with the use of GIZA++ or MGIZA.

Acknowledgments

The paper is part of the outcome of research performed under a Waseda University Grant for Special Research Project (Project number: 2015A-063).

References

Necip Fazil Ayan and Bonnie J Dorr. 2006. Going beyond AER: An extensive analysis of word alignments and their impact on MT. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 9–16. Association for Computational Linguistics.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational linguistics*, 19(1):61–74.

William A Gale and Kenneth Ward Church. 1991. Identifying word correspondences in parallel texts. In *HLT*, volume 91, pages 152–157. Citeseer.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *In Proc. of the Sixth Workshop on Statistical Machine Translation*.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952. Association for Computational Linguistics.

Philipp Koehn, Amittai Axelrod, Alexandra Birch, Chris Callison-Burch, Miles Osborne, David Talbot, and Michael White. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *IWSLT*, pages 68–75.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Adrien Lardilleux and Yves Lepage. 2009. Sampling-based multilingual alignment. In *Recent Advances in Natural Language Processing*, pages 214–218.

Adrien Lardilleux, François Yvon, and Yves Lepage. 2012. Hierarchical sub-sentential alignment with anymalign, May.

Adrien Lardilleux, François Yvon, and Yves Lepage. 2013. Generalizing sampling-based multilingual alignment. *Machine Translation*, 27(1):1–23.

Toshiaki Nakazawa, Hideya Mino, Isao Goto, Graham Neubig, Sadao Kurohashi, and Eiichiro Sumita. 2015. Overview of the 2nd workshop on asian translation. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, Kyoto, Japan, October.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics.

Baosong Yang and Yves Lepage. 2015. Leveraging the advantages of associative methods for faster training of smt systems. *Master thesis, Graduate School of Information, Production and Systems Waseda Univeristy*.

Hongyuan Zha, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu. 2001. Bipartite graph partitioning and data clustering. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 25–32. ACM.

	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 2: 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.

	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	▨															
这些		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
値		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
，		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
通过		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
upgma		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
法		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
进行		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
聚类		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
分析		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
。		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

	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	▨	▨														
这些	▨	▨														
値			▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
，			▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
通过			▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
upgma			▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
法			▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
进行			▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
聚类			▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
分析			▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
。			▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

	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	▨	▨	▨													
这些	▨	▨	▨													
値	▨	▨	▨													
，				▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
通过				▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
upgma				▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
法				▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
进行				▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
聚类				▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
分析				▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
。				▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

	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	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	
这些	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	
値	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	
，	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	
通过	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	
upgma	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	
法	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	
进行	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	
聚类	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	
分析	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	
。																▨

	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		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
这些	▨															
値	▨															
，	▨															
通过	▨															
upgma	▨															
法	▨															
进行	▨															
聚类	▨															
分析	▨															
。	▨															

	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			▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
这些			▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
値	▨	▨														
，	▨	▨														
通过	▨	▨														
upgma	▨	▨														
法	▨	▨														
进行	▨	▨														
聚类	▨	▨														
分析	▨	▨														
。	▨	▨														

	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																▨
这些																▨
値																▨
，																▨
通过																▨
upgma																▨
法																▨
进行																▨
聚类																▨
分析																▨
。	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	

Table 3: 7 out of all the possible segmentation in two blocks are shown.

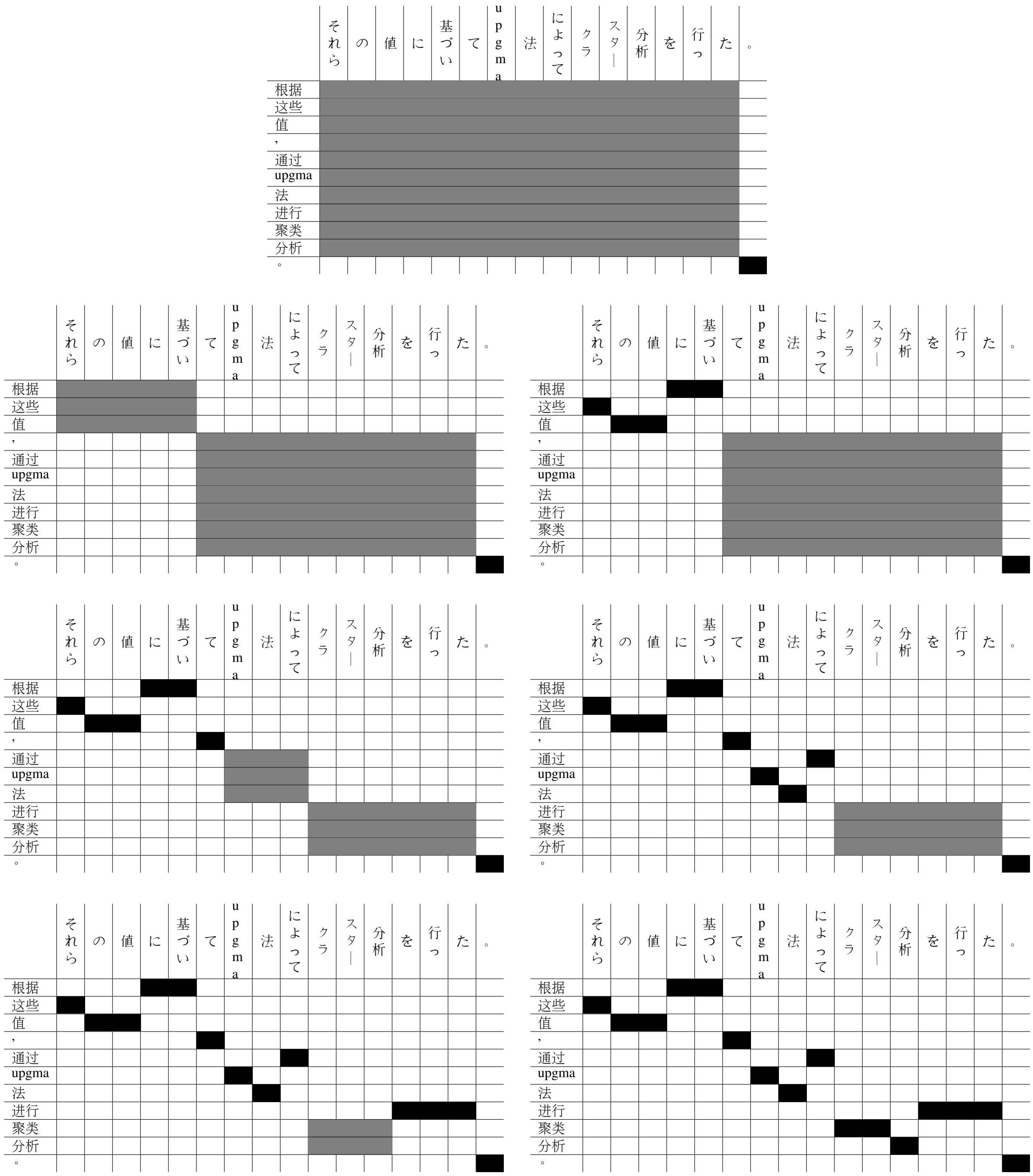

Table 4: Steps in recursive segmentation and alignment result using sampling-based alignment and hierarchical sub-sentential alignment method.

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

Table 5: Evaluation results by using different aligner (GIZA++ and MGIZA) based on the data of JPC given in Table 1.

s→t	Moses	Aligner	BLEU			RIBES			Training time
			Jum	Kyt	Mec	Jum	Kyt	Mec	
zh→ja	2.1.1	MGIZA	34.40	35.59	34.52	0.774606	0.771082	0.773494	5:34:28
	2.1.1	GIZA++	34.28	35.32	34.46	0.770829	0.767483	0.769517	4:43:56

Table 6: Evaluation results (Web server automatic evaluation) by using different aligner (GIZA++ and MGIZA) based on the data of JPC given in Table 1.

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

Table 7: Evaluation results by using the alignment method of combining sampling-based alignment and bilingual hierarchical sub-sentential alignment methods based on the data of JPC given in Table 1. In decreasing order of BLEU cores. Here, 2 (c) shows option -i of *Anymalign* is 2, and *Cutnlaign* version where core component is implemented in C.

Language	Moses	Aligner		BLEU			RIBES			Training time
		Anymalign + Cutnalign								
		Timeout (s)	i	Jum	Kyt	Mec	Jum	Kyt	Mec	
zh-ja	3.0	1200	2 (c)	33.00	33.96	33.09	0.777100	0.774241	0.776778	1:2:8
zh-ja	3.0	5400	2 (c)	32.98	33.97	33.04	0.775729	0.774008	0.774800	2:9:29
zh-ja	2.1.1	1200	2 (c)	33.24	33.70	32.95	0.771271	0.769397	0.770645	0:57:1
zh-ja	2.1.1	1200	2 (python)	33.01	33.89	33.03	0.771949	0.769415	0.770682	1:1:16

Table 8: Evaluation results (Web server automatic evaluation) by using the alignment method of combining sampling-based alignment and bilingual hierarchical sub-sentential alignment methods based on the data of JPC given in Table 1. In decreasing order of BLEU cores.