Reduction of training time in statistical machine translation: a study of the sampling-based alignment method

Meng Kong*, **Chonlathorn Kwankajornkiet**†, **Jun Li***,
Baosong Yang‡, **Lishi Zhang***, **Yujia Zhang***, **Zhongwen Zhao***
and Yves Lepage*

* Waseda University, Japan
† Chulalongkorn university, Thailand
‡ University of Macau, China

Abstract

This reports gives the results of a series of consistent experiments, the goal of which was to reduce time by using the sampling-based alignment method for the computation of word-to-word associations and the production of phrase tables. The data used are consistent across languages as we use a multilingual resource. In this way, the results may be compared across language pairs. We use two language pairs which are known to be respectively easy and difficult for statistical machine translation, and a language pair traditional in machine translation: French–English.

1. Introduction

Sampling-based multilingual alignment, introduced in (?), and implemented as `Anymalign`[1], is an associative method for the computation of word associations. The method repeatedly draws random (mainly small) sub-corpora from the parallel corpus and obtains occurrence distributions of word pairs (or short word sequence pairs) within each sub-corpus so as to ultimately produce a word association table.

Bilingual hierarchical sub-sentential alignment, introduced in (?), and implemented as `Cutnalign`[2], is an associative method to compute sub-sentential alignments. It processes parallel sentences using a recursive binary segmentation of the alignment matrix. It yields performance comparable with that of state-of-the-art methods (?).

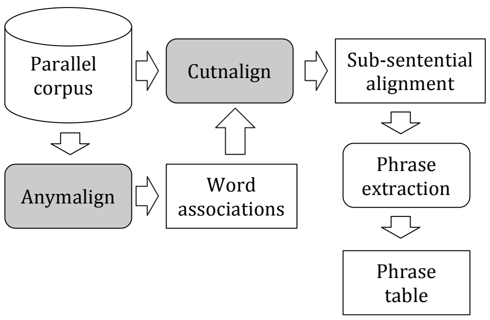

Figure 1: Combination of two associative methods, `Anymalign` and `Cutnalign`, to obtain phrase tables from a parallel corpus.

Figure 1 describes the training process which combines these two associative methods. It replaces GIZA++ and the grow-diag-final-and heuristic: `Cutnalign` uses word associations produced by `Anymalign` as input, and outputs sub-sentential alignments. The relevant script in Moses[3] then extracts phrases from sub-sentential alignments.

We present various types of improvements in the current implementations of the two above-mentioned associative methods that make them competitive with recent probabilistic approaches. The combination of the two new versions of `Anymalign` and `Cutnalign` result in an overall alignment process that can be faster than `Fast align` while delivering comparable results.

2. Multi-processing

2.1. Word associations

`Anymalign` draws random sub-corpora from the training corpus, and computes the occurrence distribution profiles for all words over all sentence pairs in each sub-corpus. Consequently, the process for each sub-corpus is independent. The sizes of the sub-corpora are randomly drawn according to a specific distribution. Consequently, sampling of sizes can also be performed independently in different sub-processes, without affecting the general behavior in any way. Multi-processing is thus done by having each sub-process randomly drawing sub-corpora sizes, drawing sub-corpora of the given sizes, and computing word associations. After the master process has received an interruption[4], word associations and their associated frequencies are output by each sub-process and passed over to the master process which sums up the frequencies of each word association produced by each sub-process and computes association scores.

Experiments show that only very small, and insignificant differences in associations output exist between the mono-processing and multi-processing versions. They are due to differences in sampling.

2.2. Hierarchical sub-sentential alignment

`Cutnalign` is easily parallelized by observing that

[1] `https://anymalign.limsi.fr/`
[2] Thanks to the authors for providing the source code.
[3] `train-model.perl --first step 4`
[4] `Anymalign` is an anytime process, and should be given a timeout.

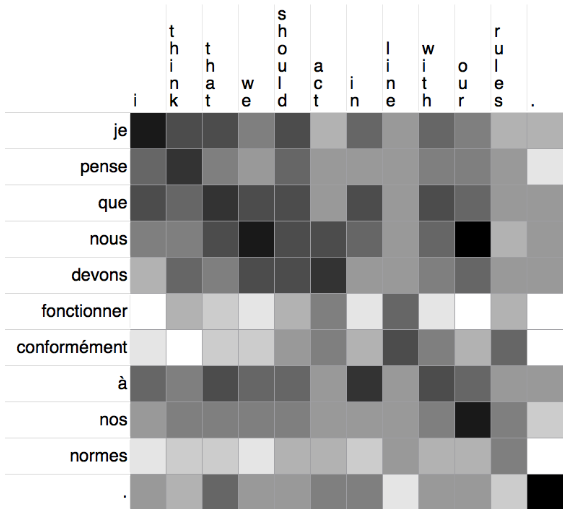

Figure 2: Translation strengths in a French–English sentence pair matrix. Cells are grayed from 0.0 (white) to 1.0 (black) on a logarithmic scale.

the sub-sentential alignment process for each different sentence pair is independent from the other ones. Experiments have shown that using 4 cores divides the time by 3.

By design, introducing multi-processing as described above does not affect the quality of the final results, because the parallelized and non-parallelized implementations are theoretically equivalent. We checked that sub-sentential alignments outputs in both implementations are exactly the same.

3. Experiments

3.1. Data

We use 3 language pairs in both directions involving 5 European languages[5]: fr–en (usual test languages), fi–en (agglutinative language–isolating language), and es–pt (close languages).

All the experiments use data from the corresponding part of the Europarl parallel corpus v3 (**?**), so that BLEU scores can be compared across language pairs, as the training, tuning and test sets correspond across languages.

Table **??** give statistics about the data. The training corpus is made of 347,614 sentences; 500 sentences are used for tuning; the test set contains 5,000 lines.

3.2. Tools

We evaluate our work by building phrase-based statistical machine translation systems basically using the Moses toolkit, lexicalized reordering models (**?**) and the KenLM Language Modeling toolkit (**?**). Accuracy relatively to translation references is assessed using BLEU.

Baselines These baselines…

MGIZA++

`Fast align`

[5]English (en), French (fr), Spanish (es), Portuguese (pt), Finnish (fi).

Language	Lines	Word tokens	Word types	Words / line
en	347,614	9.95 M	66,693	28.61
es	”	10.47 M	99,947	30.13
fi	”	7.18 M	296,954	20.66
fr	”	10.96 M	84,119	31.52
pt	”	10.29 M	102,336	29.59

(a) Training data.

Language	Lines	Word tokens	Word types	Words / line
en	500	14.61 k	2,954	29.22
es	”	15.40 k	3,495	30.80
fi	”	10.55 k	4,568	21.09
fr	”	16.16 k	3,420	32.31
pt	”	15.26 k	3,600	30.51

(b) Tuning data.

Language	Lines	Word tokens	Word types	Words / line
en	38,123	1.09 M	25,330	28.70
es	”	1.15 M	36,802	30.20
fi	”	0.79 M	84,325	20.70
fr	”	1.20 M	32,574	31.60
pt	”	1.13 M	37,570	29.64

(c) Test data.

Table 1: Statistics on the data used (k= thousand, M = million)

Anymalign alone These baselines…

- **monoprocessing version** [6] this version…
- **multiprocessing version** [7] this version…
- **multiprocessing version, with bigrams)** [8] this version…

Fast_align has no multiprocessing version. Time management is done by using options: `-t` (timeout), `-c` (number of cores used). The management of the types of alignments for `Anymalign` is done by using: `-i` (size of multi-tokens examined), `-H+NH` (hapax-oriented sampling), `-n` (minimal size of entries), `-N` (maximal size of entries), `+adhoc` (ad-hoc entries only), `+Lopez` (approximation proposed by Lopez, 2008 for the estimation of backward translation probabilities).

[6]Command: `Anymalign`

[7]Command: `Anymalign -c 4` where `-c` gives the number of cores used (here, 4).

[8]Command: `Anymalign -c 4 -i 2` where `-c` gives the number of cores used (here, 4); and `-i` gives the size of combinations of words that `Anymalign` will consider for alignment (here, 2). However, this does not imply that the maximal length of phrases is 2. It can be longer, as sequences of such combinations may be output as a phrase in the phrase table.

3.3. Machines

All experiments have been performed on HP machines. The processor is of the type i7-3770 with 4 cores, with a frequency of 3.4 GHz and memory of 16 Gbytes.

4. Conclusion

We presented a series of experiments and results obtained in the frame of a sepcail grant in aid of Waseda university. The goal of this research was to reduce time by using the sampling-based alignment method for the computation of word-to-word associations and the production of phrase tables in statistical machine translation.

5. Respective roles of the co-authors

Meng Kong, a former master student at IPS, proposed the method, and implemented the mono- and multi-processing versions of `Anymalign`, for the computation of word-to-word associations of hapaxes in one-shot and of non-hapax words by creating one or several sub-corpora per line, during his master studies at IPS.

Chonlathorn Kwankajornkiet, an undergraduate student at Chulalongkorn university, Thailand, implemented the C core component of `Cutnalign` and proposed and implemented the user-friendly interface to run experiments for the generation of ad hoc phrase tables, during a summer internship at IPS.

Jun Li, a master student at IPS, participated in the design of the new version of `Cutnalign`, and in running the majority of the baseline experiments in the six language pairs.

Baosong Yang, a former master student at IPS, proposed and implemented several improvements in `Cutnalign`, and implemented the mono- and multi-processing versions of `Anymalign` and `Cutnalign`, during his master studies at IPS.

Lishi Zhang, a former master student at IPS, proposed a new weighting scheme for the weighted sampling-based alignment method for the production of ad hoc phrase tables, implemented it, proposed and implemented a pipeline for the production of ad hoc phrase tables and participated in running experiments for the production of ad hoc phrase tables in the six language pairs, during his master studies at IPS.

Yujia Zhang, a master student at IPS, participated in running several experiments in the six language pairs.

Zhongwen Zhao, a master student at IPS, proposed and implemented a general script to run almost all the individual experiments reported in this paper, by calling various programs with various options.

Yves Lepage, a professor at IPS, was the principal investigator. He proposed several of the improvements in the methods, participated in their implementation or supervised their implementation, participated in running the experiments, synthesized the data and results and wrote the main part of the paper.

6. Acknowledgements

This unpublished paper describes part of the outcome of research performed under a Waseda University Grant for Special Research Projects (Project number: 2015A-063, title: reducing the time of development of statistical machine translation systems: a study of the sampling-based alignment method).

src	tgt	Aligner	BLEU	Training	Tuning	Decoding
					Times (h:m)	
es	pt	MGIZA++	36.9 ± 0.2	2:30	3:	11:30
es	pt	Fast align	36.9 ± 0.2	1:	1:30	10:
pt	es	MGIZA++	39.2 ± 0.2	2:30	2:30	9:30
pt	es	Fast align	38.9 ± 0.2	1:	2:	9:
en	fr	MGIZA++	40.0 ± 0.2	2:30	3:	10:30
en	fr	Fast align	39.7 ± 0.2	:46	2:30	20:
fr	en	MGIZA++	34.7 ± 0.2	3:	2:30	11:30
fr	en	Fast align	34.6 ± 0.2	:44	1:	11:
fi	en	MGIZA++	26.5 ± 0.2	2:	2:30	4:30
fi	en	Fast align	26.4 ± 0.2	:40	:25	4:30
en	fi	MGIZA++	16.4 ± 0.2	2:	3:30	9:
en	fi	Fast align	16.4 ± 0.2	:38	2:	9:

Table 2: Baseline results for all language pairs

Alignment method	# of cores	Options for `Anymalign`	BLEU score	Training	Tuning	Decoding
					Times (min)	
MGIZA++	4					
`Fast align`	1[a]			t_{fa} = xx		
`Anymalign`	1			t_{fa}		
`Anymalign`	4			t_{fa}		
`Anymalign`	4	`-i 2`		t_{fa}		
`Anymalign`	1	`-H+NH -i 2`				
`Anymalign`	4	`-H+NH -i 2`				
`Anymalign`	4	`-H+NH -i 2`		t_{fa}		
`Anymalign+Cutnalign`	4			t_{fa}		
`Anymalign+Cutnalign`	4	`-i 2`		t_{fa}		
`Anymalign+Cutnalign`	4	`-H+NH`		t_{fa}		
`Anymalign`	4	`-adhoc -Lopez`				
`Anymalign`	4	`-adhoc`				

(a) Results for the Spanish-Portuguese language pair

[a] `Fast align` has no mutli-processing version.

Word-to-word associations	Options	Sub-sentential alignment	Options	BLEU	Times (mn) Training	Tuning	Decoding
MGIZA++		grow-diag-final		39.20 ± 0.20	150	150	570
Fast_align		grow-diag-final		38.97 ± 0.19	$t_{fa} = 53$	144	548
Anymalign	+adhoc	grow-diag-final		36.83 ± 0.21	56	123	424
Anymalign	+adhoc +Lopez	grow-diag-final		36.88 ± 0.20	58	147	407
Anymalign	-t ($t_{fa}-2$mn) -c 4	*None*		*Experiment not performed*			
Anymalign	-t ($t_{fa}-2$mn) -c 4 -n 1 -N 1	*None*		*Experiment not performed*			
Anymalign	-t t_{fa} -c 4 -i 2	*None*		36.05 ± 0.19	$57 > t_{fa}$	122	409
Anymalign	-t ($t_{fa}-2$mn) -c 4 -H+NH	*None*		36.12 ± 0.21	$60 > t_{fa}$	101	415
Anymalign	-t ($t_{fa}-2$mn) -c 4	Cutnalign	-c 4	38.74 ± 0.20	t_{fa}	180	635
Anymalign	-t ($t_{fa}-2$mn) -c 4 -n 1 -N 1	Cutnalign	-c 4	38.67 ± 0.21	t_{fa}	240	752
Anymalign	-t ($t_{fa}-2$mn) -c 4 -i 2	Cutnalign	-c 4	38.86 ± 0.20	t_{fa}	96	533
Anymalign	-t ($t_{fa}-2$mn) -c 4 -H+NH	Cutnalign	-c 4	38.88 ± 0.21	$59 > t_{fa}$	67	588

Table 4: All results for the Portuguese–Spanish language pair. The version of `Anymalign` with option `-H+NH` takes more time than `Fast_align`.

Word-to-word associations	Options	Sub-sentential alignment	Options	BLEU	Times (mn)		
					Training	Tuning	Decoding
MGIZA++		grow-diag-final		36.90 ± 0.20	150	180	690
Fast_align		grow-diag-final		36.74 ± 0.20	t_{fa} = 50	138	612
Anymalign	+adhoc	grow-diag-final		35.74 ± 0.21	233	112	395
Anymalign	+adhoc +Lopez	grow-diag-final		35.80 ± 0.20	185	113	405
Anymalign	-t ($t_{fa}-2$mn) -c 4	*None*		*Experiment not performed*			
Anymalign	-t ($t_{fa}-2$mn) -c 4 -n 1 -N 1	*None*		*Experiment not performed*			
Anymalign	-t t_{fa} -c 4 -i 2	*None*		34.28 ± 0.19	t_{fa}	51	435
Anymalign	-t ($t_{fa}-2$mn) -c 4 -H+NH	*None*		34.62 ± 0.20	$54 > t_{fa}$	115	439
Anymalign	-t ($t_{fa}-2$mn) -c 4	Cutnalign	-c 4	36.56 ± 0.20	t_{fa}	138	612
Anymalign	-t ($t_{fa}-2$mn) -c 4 -n 1 -N 1	Cutnalign	-c 4	36.53 ± 0.20	t_{fa}	148	525
Anymalign	-t ($t_{fa}-2$mn) -c 4 -i 2	Cutnalign	-c 4	36.58 ± 0.20	t_{fa}	180	540
Anymalign	-t ($t_{fa}-2$mn) -c 4 -H+NH	Cutnalign	-c 4	36.70 ± 0.21	$53 > t_{fa}$	165	600

Table 5: All results for the Spanish–Portuguese language pair. The version of `Anymalign` with option `-H+NH` takes more time than `Fast_align`.

Word-to-word associations	Options	Sub-sentential alignment	Options	BLEU	Times (mn) Training	Tuning	Decoding
MGIZA++		grow-diag-final		34.70 ± 0.20	180	150	690
Fast_align		grow-diag-final		34.59 ± 0.21	$t_{fa} = 48$	62	648
Anymalign	+adhoc	grow-diag-final		30.71 ± 0.20	223	150	497
Anymalign	+adhoc +Lopez	grow-diag-final		30.67 ± 0.20	172	136	555
Anymalign	-t (t_{fa}−2mn) -c 4	*None*		*Experiment not performed*			
Anymalign	-t (t_{fa}−2mn) -c 4 -n 1 -N 1	*None*		*Experiment not performed*			
Anymalign	-t t_{fa} -c 4 -i 2	*None*		28.97 ± 0.20	t_{fa}	52	478
Anymalign	-t (t_{fa}−2mn) -c 4 -H+NH	*None*		29.12 ± 0.19	$51 > t_{fa}$	54	480
Anymalign	-t (t_{fa}−2mn) -c 4	Cutnalign	-c 4	33.68 ± 0.20	t_{fa}	137	698
Anymalign	-t (t_{fa}−2mn) -c 4 -n 1 -N 1	Cutnalign	-c 4	33.86 ± 0.21	t_{fa}	191	570
Anymalign	-t (t_{fa}−2mn) -c 4 -i 2	Cutnalign	-c 4	33.96 ± 0.20	t_{fa}	123	560
Anymalign	-t (t_{fa}−2mn) -c 4 -H+NH	Cutnalign	-c 4	34.14 ± 0.22	$45 < t_{fa}$	138	649

Table 6: All results for the French–English language pair. The version of `Anymalign` with option `-H+NH` halts before the timeout is reached.

Word-to-word associations	Options	Sub-sentential alignment	Options	BLEU	Times (mn) Training	Tuning	Decoding
MGIZA++		grow-diag-final		40.00 ± 0.20	150	180	630
Fast_align		grow-diag-final		39.64 ± 0.17	$t_{fa} = 46$	142	845
Anymalign	+adhoc	grow-diag-final		36.06 ± 0.20	243	134	557
Anymalign	+adhoc +Lopez	grow-diag-final		36.11 ± 0.21	198	111	523
Anymalign	-t (t_{fa}−2mn) -c 4	*None*		*Experiment not performed*			
Anymalign	-t (t_{fa}−2mn) -c 4 -n 1 -N 1	*None*		*Experiment not performed*			
Anymalign	-t t_{fa} -c 4 -i 2	*None*		35.69 ± 0.20	$49 > t_{fa}$	127	430
Anymalign	-t (t_{fa}−2mn) -c 4 -H+NH	*None*		35.83 ± 0.19	$51 > t_{fa}$	115	426
Anymalign	-t (t_{fa}−2mn) -c 4	Cutnalign	-c 4	38.67 ± 0.20	t_{fa}	87	537
Anymalign	-t (t_{fa}−2mn) -c 4 -n 1 -N 1	Cutnalign	-c 4	38.61 ± 0.21	t_{fa}	180	540
Anymalign	-t (t_{fa}−2mn) -c 4 -i 2	Cutnalign	-c 4	39.05 ± 0.20	t_{fa}	93	574
Anymalign	-t (t_{fa}−2mn) -c 4 -H+NH	Cutnalign	-c 4	38.94 ± 0.21	$55 > t_{fa}$	120	536

Table 7: All results for the English–French language pair. The version of `Anymalign` with option `-H+NH` takes more time than `Fast_align`.

Word-to-word associations	Options	Sub-sentential alignment	Options	BLEU	Times (mn)		
					Training	Tuning	Decoding
MGIZA++		grow-diag-final		26.50 ± 0.20	120	150	270
Fast_align		grow-diag-final		26.39 ± 0.21	$t_{fa} = 48$	38	291
Anymalign	+adhoc	grow-diag-final		20.12 ± 0.20	225	123	248
Anymalign	+adhoc +Lopez	grow-diag-final		20.86 ± 0.20	173	106	343
Anymalign	-t ($t_{fa}-2$mn) -c 4	*None*		*Experiment not performed*			
Anymalign	-t ($t_{fa}-2$mn) -c 4 -n 1 -N 1	*None*		*Experiment not performed*			
Anymalign	-t ($t_{fa}-2$mn) -c 4 -i 2	*None*		20.12 ± 0.20	t_{fa}	100	182
Anymalign	-t ($t_{fa}-2$mn) -c 4 -H+NH	*None*		20.86 ± 0.20	$37 < t_{fa}$	82	193
Anymalign	-t ($t_{fa}-2$mn) -c 4	Cutnalign	-c 4	23.80 ± 0.19	t_{fa}	71	255
Anymalign	-t ($t_{fa}-2$mn) -c 4 -n 1 -N 1	Cutnalign	-c 4	23.87 ± 0.19	t_{fa}	57	257
Anymalign	-t ($t_{fa}-2$mn) -c 4 -i 2	Cutnalign	-c 4	24.53 ± 0.19	t_{fa}	42	240
Anymalign	-t ($t_{fa}-2$mn) -c 4 -H+NH	Cutnalign	-c 4	24.23 ± 0.19	$39 < t_{fa}$	81	234

Table 8: All results for the Finnish–English language pair. The version of `Anymalign` with option `-H+NH` halts before the timeout is reached.

Word-to-word associations	Options	Sub-sentential alignment	Options	BLEU	Times (mn) Training	Tuning	Decoding
MGIZA++		grow-diag-final		16.40 ± 0.20	120	210	540
Fast_align		grow-diag-final		16.42 ± 0.17	$t_{fa} = 37$	71	547
Anymalign	+adhoc	grow-diag-final		12.69 ± 0.20	194	70	453
Anymalign	+adhoc +Lopez	grow-diag-final		12.88 ± 0.20	186	144	588
Anymalign	-t (t_{fa}−2mn) -c 4	*None*		*Experiment not performed*			
Anymalign	-t (t_{fa}−2mn) -c 4 -n 1 -N 1	*None*		*Experiment not performed*			
Anymalign	-t t_{fa} -c 4 -i 2	*None*		12.41 ± 0.15	t_{fa}	106	579
Anymalign	-t (t_{fa}−2mn) -c 4 -H+NH	*None*		12.27 ± 0.15	$28 < t_{fa}$	150	389
Anymalign	-t (t_{fa}−2mn) -c 4	Cutnalign	-c 4	15.64 ± 0.18	t_{fa}	103	429
Anymalign	-t (t_{fa}−2mn) -c 4 -n 1 -N 1	Cutnalign	-c 4	15.58 ± 0.19	t_{fa}	115	474
Anymalign	-t (t_{fa}−2mn) -c 4 -i 2	Cutnalign	-c 4	15.88 ± 0.18	t_{fa}	147	499
Anymalign	-t (t_{fa}−2mn) -c 4 -H+NH	Cutnalign	-c 4	15.73 ± 0.17	$30 < t_{fa}$	120	496

Table 9: All results for the English–Finnish language pair. The version of `Anymalign` with option `-H+NH` halts before the timeout is reached.