

# Lexicons or phrase tables? An investigation in sampling-based multilingual alignment

Adrien Lardilleux\* Jonathan Chevelu<sup>†</sup>\* Yves Lepage\*  
Ghislain Putois<sup>†</sup> Julien Gosme\*

\*GREYC, Université de Caen Basse-Normandie  
<sup>†</sup>Orange Labs, Lannion  
France

12th November 2009

# Outline

What is sampling-based multilingual alignment?

Some typical results on two typical tasks

What is in the phrase tables?

What is missing?

## A sub-sentential alignment method

`anymalign.py`

Freely available, open source, easy to use, portable, pythonic

Extract lexical equivalences from sentence-aligned parallel corpora:

**Multiword:** extract translations of (dis)contiguous sequences of words

**Multilingual:** can process any number of languages at a time

**“Anytime”:** quality is not a matter of time.

Coverage is a matter of time.

**Simple:** very simple

# An example-based sub-sentential alignment method

Alignments detection:

based on **strict distribution similarities** of words  
on a multilingual parallel corpus

Alignments extraction:

based on **string differences**

Alignments scoring:

straightforward statistics

## An example (1/3: strict distribution similarities)

Input: a subcorpus obtained by **sampling** the initial training corpus

1		One <sub>1</sub> coffee <sub>1</sub> , <sub>1</sub> please <sub>1</sub> . <sub>1</sub> Un <sub>2</sub> café <sub>2</sub> , <sub>2</sub> s'il <sub>2</sub> vous <sub>2</sub> plaît <sub>2</sub> . <sub>2</sub>
2		This <sub>1</sub> coffee <sub>1</sub> is <sub>1</sub> excellent <sub>1</sub> . <sub>1</sub> Ce <sub>2</sub> café <sub>2</sub> n'est <sub>2</sub> pas <sub>2</sub> mauvais <sub>2</sub> . <sub>2</sub>
3		One <sub>1</sub> strong <sub>1</sub> tea <sub>1</sub> . <sub>1</sub> Un <sub>2</sub> thé <sub>2</sub> fort <sub>2</sub> . <sub>2</sub>

## An example (1/3: strict distribution similarities)

Input: a subcorpus obtained by **sampling** the initial training corpus

1	One <sub>1</sub> coffee <sub>1</sub> , <sub>1</sub> please <sub>1</sub> . <sub>1</sub> Un <sub>2</sub> café <sub>2</sub> , <sub>2</sub> s'il <sub>2</sub> vous <sub>2</sub> plaît <sub>2</sub> . <sub>2</sub>
2	This <sub>1</sub> coffee <sub>1</sub> is <sub>1</sub> excellent <sub>1</sub> . <sub>1</sub> Ce <sub>2</sub> café <sub>2</sub> n'est <sub>2</sub> pas <sub>2</sub> mauvais <sub>2</sub> . <sub>2</sub>
3	One <sub>1</sub> strong <sub>1</sub> tea <sub>1</sub> . <sub>1</sub> Un <sub>2</sub> thé <sub>2</sub> fort <sub>2</sub> . <sub>2</sub>



	. <sub>1</sub>	. <sub>2</sub>	. <sub>1</sub>	. <sub>2</sub>	Ce <sub>2</sub>	One <sub>1</sub>	This <sub>1</sub>	Un <sub>2</sub>	café <sub>2</sub>	coffee <sub>1</sub>	excellent <sub>1</sub>	fort <sub>2</sub>	is <sub>1</sub>	mauvais <sub>2</sub>	n'est <sub>2</sub>	pas <sub>2</sub>	plaît <sub>2</sub>	please <sub>1</sub>	s'il <sub>2</sub>	strong <sub>1</sub>	tea <sub>1</sub>	thé <sub>2</sub>	vous <sub>2</sub>
1	1	1	1	1	0	1	0	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0	1
2	0	0	1	1	1	0	1	0	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0
3	0	0	1	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0

## An example (1/3: strict distribution similarities)

Input: a subcorpus obtained by **sampling** the initial training corpus

1	One <sub>1</sub> coffee <sub>1</sub> , <sub>1</sub> please <sub>1</sub> . <sub>1</sub> Un <sub>2</sub> café <sub>2</sub> , <sub>2</sub> s'il <sub>2</sub> vous <sub>2</sub> plaît <sub>2</sub> . <sub>2</sub>
2	This <sub>1</sub> coffee <sub>1</sub> is <sub>1</sub> excellent <sub>1</sub> . <sub>1</sub> Ce <sub>2</sub> café <sub>2</sub> n'est <sub>2</sub> pas <sub>2</sub> mauvais <sub>2</sub> . <sub>2</sub>
3	One <sub>1</sub> strong <sub>1</sub> tea <sub>1</sub> . <sub>1</sub> Un <sub>2</sub> thé <sub>2</sub> fort <sub>2</sub> . <sub>2</sub>



	. <sub>1</sub>	. <sub>2</sub>	. <sub>1</sub>	. <sub>2</sub>	Ce <sub>2</sub>	One <sub>1</sub>	This <sub>1</sub>	Un <sub>2</sub>	café <sub>2</sub>	coffee <sub>1</sub>	excellent <sub>1</sub>	fort <sub>2</sub>	is <sub>1</sub>	mauvais <sub>2</sub>	n'est <sub>2</sub>	pas <sub>2</sub>	plaît <sub>2</sub>	please <sub>1</sub>	s'il <sub>2</sub>	strong <sub>1</sub>	tea <sub>1</sub>	thé <sub>2</sub>	vous <sub>2</sub>
1	1	1	1	1	0	1	0	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0	1
2	0	0	1	1	1	0	1	0	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0
3	0	0	1	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0



	. <sub>1</sub>	. <sub>2</sub>	café <sub>2</sub>	coffee <sub>1</sub>	One <sub>1</sub>	Un <sub>2</sub>	. <sub>1</sub>	. <sub>2</sub>	plaît <sub>2</sub>	please <sub>1</sub>	s'il <sub>2</sub>	vous <sub>2</sub>	Ce <sub>2</sub>	This <sub>1</sub>	excellent <sub>1</sub>	is <sub>1</sub>	mauvais <sub>2</sub>	n'est <sub>2</sub>	pas <sub>2</sub>	fort <sub>2</sub>	strong <sub>1</sub>	tea <sub>1</sub>	thé <sub>2</sub>
1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0
3	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

## An example (2/3: string differences)

The words:	appear on lines:	from which we extract:
coffee <sub>1</sub> café <sub>2</sub>	1	coffee <sub>1</sub> café <sub>2</sub> One <sub>1</sub> - , <sub>1</sub> please <sub>1</sub> . <sub>1</sub> Un <sub>2</sub> - , <sub>2</sub> s'il <sub>2</sub> vous <sub>2</sub> plaît <sub>2</sub> . <sub>2</sub>
	2	coffee <sub>1</sub> café <sub>2</sub> This <sub>1</sub> - is <sub>1</sub> excellent <sub>1</sub> . <sub>1</sub> Ce <sub>2</sub> - n'est <sub>2</sub> pas <sub>2</sub> mauvais <sub>2</sub> . <sub>2</sub>
⋮	⋮	⋮



## An example (2/3: string differences)

The words:	appear on lines:	from which we extract:
coffee <sub>1</sub> café <sub>2</sub>	1	coffee <sub>1</sub> café <sub>2</sub> One <sub>1</sub> _ , <sub>1</sub> please <sub>1</sub> . <sub>1</sub> Un <sub>2</sub> _ , <sub>2</sub> s'il <sub>2</sub> vous <sub>2</sub> plaît <sub>2</sub> . <sub>2</sub>
	2	coffee <sub>1</sub> café <sub>2</sub> This <sub>1</sub> _ is <sub>1</sub> excellent <sub>1</sub> . <sub>1</sub> Ce <sub>2</sub> _ n'est <sub>2</sub> pas <sub>2</sub> mauvais <sub>2</sub> . <sub>2</sub>
⋮	⋮	⋮



English		French	Count
coffee	↔	café	2
One _ , please .	↔	Un _ , s'il vous plaît .	1
This _ is excellent .	↔	Ce _ n'est pas mauvais .	1
		⋮	⋮

## An example (3/3: score alignments)

- ▶ The same process is repeated for numerous random subcorpora.
- ▶ All alignments from all subcorpora are collected.
- ▶ Translation probabilities are computed based on alignments' counts.

### Result:

A full-fledged translation table.

## An example (3/3: score alignments)

- ▶ The same process is repeated for numerous random subcorpora.
- ▶ All alignments from all subcorpora are collected.
- ▶ Translation probabilities are computed based on alignments' counts.

### Result:

A full-fledged translation table.

... or not?

What is sampling-based multilingual alignment?

Some typical results on two typical tasks

What is in the phrase tables?

What is missing?

## Two typical tasks

1. A machine translation task
2. A bilingual lexicon induction task

We compare the outputs of two word aligners:

1. Anymalign
2. MGIZA++, augmented by Moses for symmetric alignment and phrase extraction and scoring

We use two bilingual parallel corpora of different natures:

1. 40,000 pairs of Japanese-English sentences from the BTEC (average sentence length: 10 words)
2. 200,000 pairs of French-English sentences from the Europarl corpus (average sentence length: 31 words)

## Evaluation 1: a machine translation task

Using the Moses phrase-based SMT decoder

BTEC: short Japanese-English sentences

Phrase table origin	BLEU	TER
Anymalign	<b>0.39</b>	<b>0.45</b>
MGIZA++/Moses	0.38	<b>0.45</b>

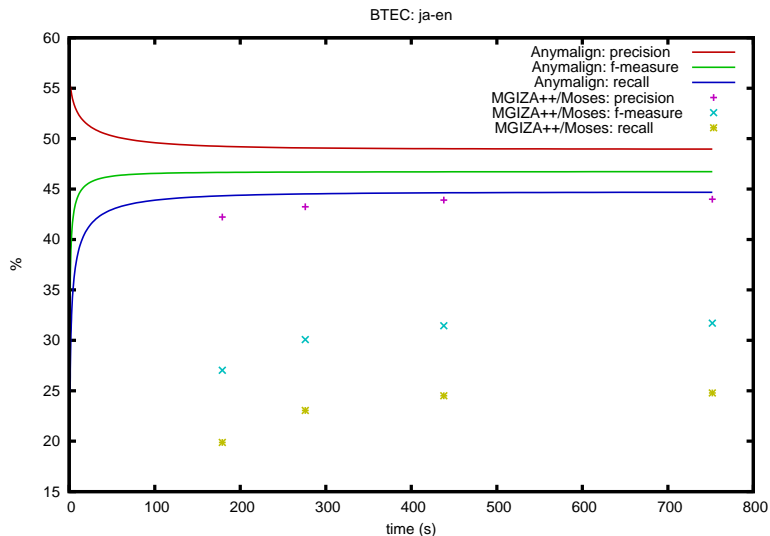
Europarl: long French-English sentences

Phrase table origin	BLEU	TER
Anymalign	0.25	0.60
MGIZA++/Moses	<b>0.29</b>	<b>0.56</b>

## Evaluation 2: a bilingual lexicon induction task (1/3)

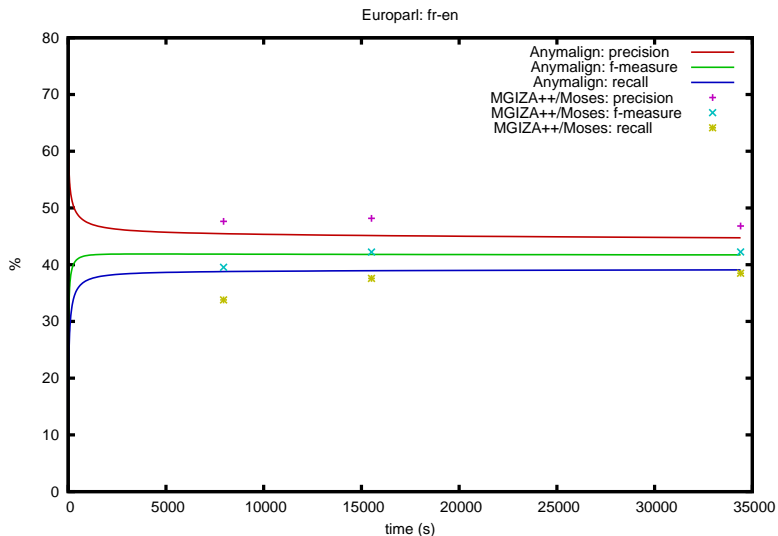
- ▶ We compare the phrase tables to a reference bilingual lexicon.
- ▶ The reference bilingual lexicon is filtered so that it contains only translation pairs that can actually be obtained from the training parallel corpus.
- ▶ We compute precision, recall, and f-measure. Translation pairs from the phrase tables are weighted according to their source-to-target translation probabilities.

# Evaluation 2: a bilingual lexicon induction task (2/3)





# Evaluation 2: a bilingual lexicon induction task (3/3)



## Conclusion of the two experiments

Anymalign typically yields equal or **worse** results  
on **phrase-based machine translation** tasks

+

Anymalign typically yields equal or **better** results  
on **bilingual lexicon induction** tasks,  
involving mainly unigrams

## Conclusion of the two experiments

Anymalign typically yields equal or **worse** results  
on **phrase-based machine translation** tasks

+

Anymalign typically yields equal or **better** results  
on **bilingual lexicon induction** tasks,  
involving mainly unigrams

=

Aren't we just aligning unigrams, and missing longer n-grams?

What is sampling-based multilingual alignment?

Some typical results on two typical tasks

What is in the phrase tables?

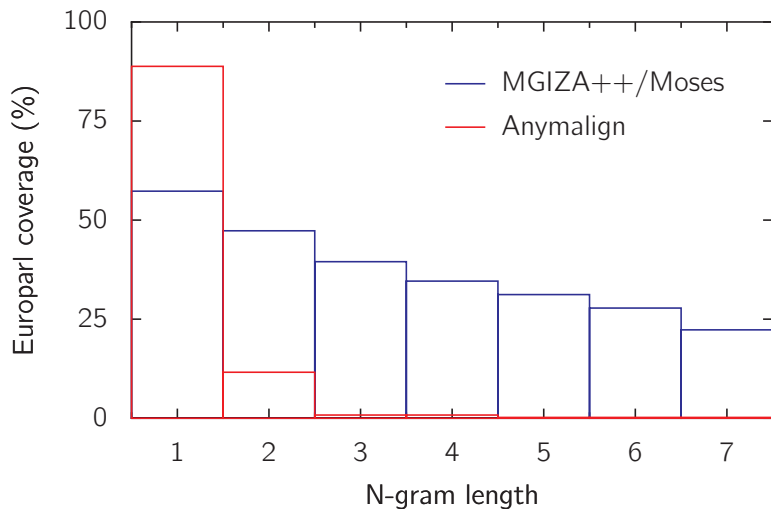
What is missing?

## Investigating the contents of alignments: settings

We now resort to 1,000,000 pairs of French-English sentences from the Europarl corpus.

- ▶ We obtained the worst results on this corpus in the previous experiments.
- ▶ A large training corpus will highlight differences between the two phrase tables.

# Investigating phrase table coverage



## Less data is worse data

Anymalign's phrase table is **42 times smaller** than MGIZA++/Moses'!

- ▶ Anymalign is much better at unigram extraction.
- ▶ Anymalign is much much much worse at n-gram extraction ( $n \geq 2$ ).

⇒ Quantity, not quality!

## Failing at aligning n-grams?

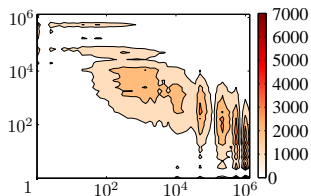
Manual inspection of the content of phrase tables suggests that Anymalign would not align sequences of words with **different frequencies**.

⇒ We plot the distribution of bigrams according to the frequency of the words they are made of.

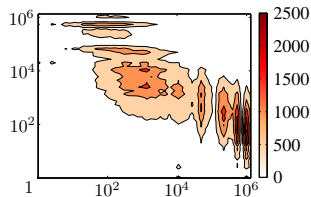


## Investigating bigrams distribution

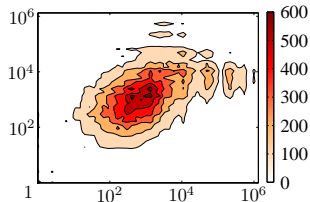
Europarl



MGIZA++/Moses

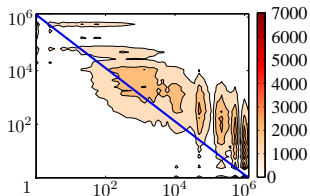


Anymalign

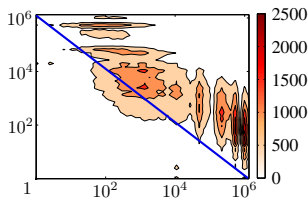


## Investigating bigrams distribution

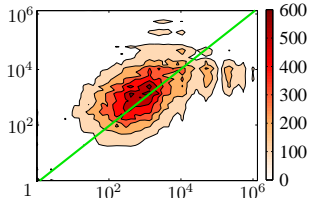
Europarl



MGIZA++/Moses



Anymalign



## But why?

### Basics of the method:

Extract sequences of words that share exactly the same distribution in a subcorpus.

Words with very different frequencies never share the same distribution, whatever the subcorpus!

## But why?

### Basics of the method:

Extract sequences of words that share exactly the same distribution in a subcorpus.

Words with very different frequencies never share the same distribution, whatever the subcorpus!

### From this corpus:

1		a b ?	α β ;
2		a c ?	α γ ;
3		d ?	δ ;

## But why?

### Basics of the method:

Extract sequences of words that share exactly the same distribution in a subcorpus.

Words with very different frequencies never share the same distribution, whatever the subcorpus!

From this corpus:

1		a b ?	α β ;
2		a c ?	α γ ;
3		d ?	δ ;

we can extract:

b	↔	β
?	↔	;
:		

## But why?

### Basics of the method:

Extract sequences of words that share exactly the same distribution in a subcorpus.

Words with very different frequencies never share the same distribution, whatever the subcorpus!

From this corpus:

1		a b ?	α β ;
2		a c ?	α γ ;
3		d ?	δ ;

we can extract:

$$b \leftrightarrow \beta$$

$$? \leftrightarrow ;$$

$$\vdots$$

but we cannot extract:

$$b ? \leftrightarrow \beta ;$$

What is sampling-based multilingual alignment?

Some typical results on two typical tasks

What is in the phrase tables?

What is missing?

## What remains to be done

Recombine alignments together in order to produce longer alignments:

Alignments known

$$\begin{array}{l} b \leftrightarrow \beta \\ ? \leftrightarrow ; \\ a \ b \ ? \leftrightarrow \alpha \ \beta \ ; \end{array}$$

$\Rightarrow$

New alignment

$$b \ ? \leftrightarrow \beta \ ;$$

- $\simeq$  extract phrase alignments consistent with word alignments
- $\simeq$  phrase extraction for phrase-based SMT



## Conclusion

- ▶ An example-based sub-sentential alignment method
- ▶ Better results on lexicon induction tasks than on MT tasks  
⇒ better at unigram extraction
- ▶ Does not align together words with different frequencies
- ▶ We would just need to recombine word alignments together in order to produce longer alignments

`anymalign.py`

`http://users.info.unicaen.fr/~alardill/anymalign/`

## Conclusion

- ▶ An example-based sub-sentential alignment method
- ▶ Better results on lexicon induction tasks than on MT tasks  
⇒ better at unigram extraction
- ▶ Does not align together words with different frequencies
- ▶ We would just need to recombine word alignments together in order to produce longer alignments

`anymalign.py`

`http://users.info.unicaen.fr/~alardill/anymalign/`

Thank you!