The GREYC Machine Translation System for the IWSLT 2008 campaign

Yves Lepage   Adrien Lardilleux   Julien Gosme   Jean-Luc Manguin

GREYC, University of Caen, France

The system

- Evolution of the ALEPH machine translation system that participated in the IWSLT 2005 [Lepage & Denoual, 2005] and IWSLT 2007 [Lepage & Lardilleux, 2007] campaigns.
- ALEPH is a pure example-based system that exploits proportional analogies (analogies of form).

Previous system: analogies between character strings:

you swim : he swims :: you surf :: he surfs

New system: can also work on words (used in IWSLT):

My hotel sucks : Your hotel sucks :: My hotel rocks :: Your hotel rocks

⇒ Nothing the character-based approach cannot deal with, but faster.

The participation of the GREYC

Tracks: all btec tasks

- Arabic to English
- Chinese to English
- Chinese to Spanish
- Chinese to Spanish by the way of English (Pivot)

Conditions: used only training data (no development set)

Non-deterministic analogy solver

Previous implementation in C:

$$x : y :: z : ? \qquad \Rightarrow \qquad ? = t$$

New solver in Python:

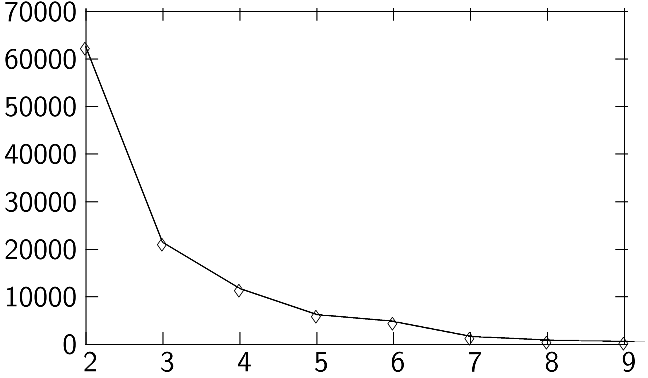
$$x : y :: z : ? \qquad \Rightarrow \qquad ? = \begin{array}{l} t_1 \\ t_2 \\ t_3 \\ \vdots \end{array}$$

Example

kalb : kulaib :: masjid : musaijid
kalb : kulaib :: masjid : musjaiid
kalb : kulaib :: masjid : musjiaid

Non-deterministic analogy solver

Distribution of the number of analogical equations
with the same number of solutions
(number of solutions in abscissae; number of analogies in ordinates):



Ratio 1 solution:multiple solutions = 30:1

Re-engineering of the engine

Main issue of the engine

Efficient discovery of translation examples that are likely to form an analogical equation is critical.

$\Rightarrow$ Design of a new heuristic:

- Analogical terms are chosen according to their longest common substring.
- Can be pre-computed and saved on disk to speed up searches.

Benefit: number of attempted analogical equations that have at least one solution increased from 28% to 52%.

New alignment method

"Perfect" alignments contain those words that strictly appear on the same lines:

Allons boire un verre .	↔	Let 's have a drink .
Allons boire une bière ou deux .	↔	Let 's have a beer or two .
Une bière et un café .	↔	One beer and one coffee .
Je voudrais un verre de vin , s' il vous plaît .	↔	I 'd like a glass of wine , please .
Je voudrais de la bière , s' il vous plaît .	↔	I 'd like some beer , please .
Nous prendrons un pichet de vin .	↔	We 'll have a jug of wine .

"Perfect"

bière ↔ beer

⇒

Contexts

Allons boire une _ ou deux .	↔	Let 's have a _ or two .
Une _ et un café .	↔	One _ and one coffee .
Je voudrais de la _ , s' il vous plaît .	↔	I 'd like some _ , please .

"Perfect"

Je voudrais _ , s' il vous plaît

↔ I 'd like _ , please

⇒

Contexts

un verre de vin _ .	↔	a glass of wine _ .
de la bière _ .	↔	some beer _ .

New alignment method

How to extract the alignments for ambiguous terms?

Allons boire un verre .	↔	Let 's have a drink .
Allons boire une bière ou deux .	↔	Let 's have a beer or two .
Une bière et un café .	↔	One beer and one coffee .
Je voudrais un verre de vin , s' il vous plaît .	↔	I 'd like a glass of wine , please .
Je voudrais de la bière , s' il vous plaît .	↔	I 'd like some beer , please .
Nous prendrons un pichet de vin .	↔	We 'll have a jug of wine .

Make them perfect: split the corpus.

"Perfect"

verre ↔ drink

⇒

Contexts

Allons boire un _ . ↔ Let 's have a _ .

"Perfect"

verre ↔ glass

⇒

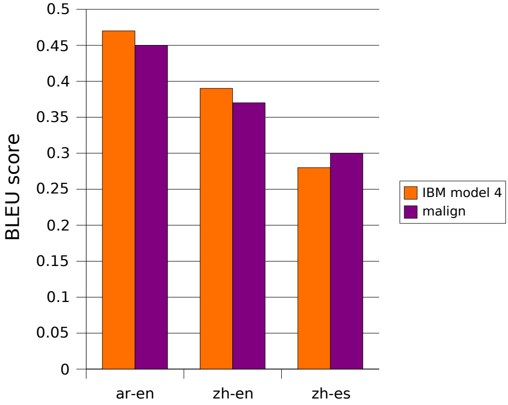
Contexts

Je voudrais un _ de vin , s' il vous plaît . ↔ I 'd like a _ of wine , please .

$$P(\text{drink}|\text{verre}) = 0.5 \qquad P(\text{verre}|\text{drink}) = 1$$
$$P(\text{glass}|\text{verre}) = 0.5 \qquad P(\text{verre}|\text{glass}) = 1$$

New alignment method

Experiments on development set 3, using the first half for tuning and the second half for testing:



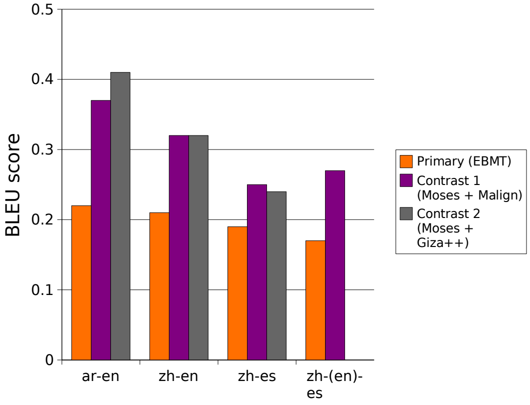
Details of the runs

3 runs for each task:

- **Primary:** ALEPH (EBMT), with training data inflated with alignments generated by malign [Lardillleux & Lepage, next Wednesday];
- **Contrast 1:** Moses [Koehn et al., 2007] with translation tables generated by malign;
- **Contrast 2:** Moses with default translation tables (refined alignments from IBM model 4, with Giza++ [Och & Ney, 2003]).

Evaluation results

Results with CRR, case+punc:



Results synthesis

In most cases: $\textit{primary} < \textit{contrast}1 \leq \textit{contrast}2$

- If one sees the *contrast*2 as a kind of baseline,
  then our system could not even reach the baseline of SMT
  in its current state (recursivity not ready at the time of evaluation).
- + only training data was used... (and you?)
- There is room for improvement!