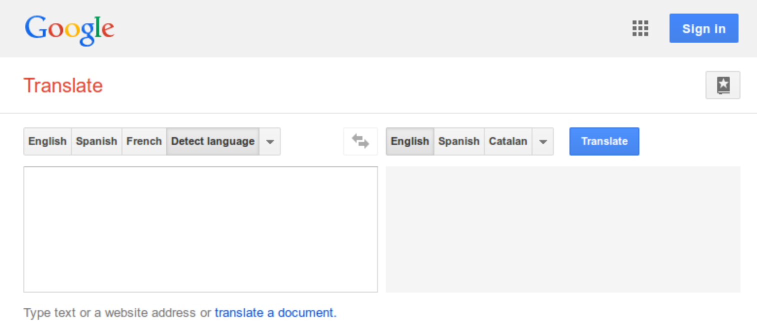
Statistical Machine Translation: Main Components

Cristina España i Bonet
DFKI GmbH

1er. Congreso Internacional de
Procesamiento de Lenguaje Natural para Lenguas Indígenas

Morelia, Michoacán, México
5th November, 2020

Neural Machine Translation (NMT), SotA in everyday MT



RBMT vs. SMT vs. NMT for High-Quality Systems

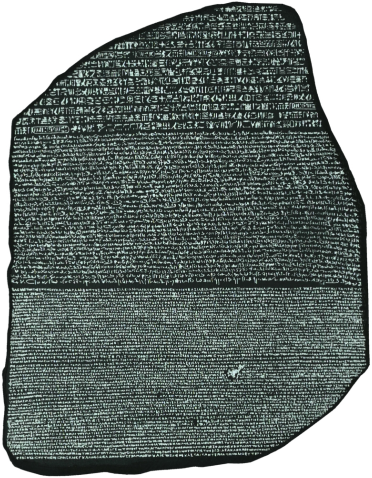
	RBMT	**SMT**	**NMT**
Data Amount	small	medium	large
Training Time	–	days	weeks
CPU/GPU	CPU	CPU	GPU
Cost	expensive (in people)	cheap	expensive (in hardware)
Maintainability	weak	strong	superstrong
Grammaticality	strong	medium	strong
Reordering	strong	weak	strong
Consistency	strong	medium	weak
Coverage	weak	strong	weak
Multilinguality	medium	none	strong

Today's Goal: Understand SMT via Moses

```
echo 'das ist ein kleines haus' | moses -f moses.ini
```

Outline


Empirical MT
relies on
aligned
corpora



Aligned parallel corpora: Numbers

Corpora

Corpus	# segments (app.)	# words (app.)
JRC-Acquis	$1.0 \cdot 10^6$	$30 \cdot 10^6$
Europarl	$2.0 \cdot 10^6$	$55 \cdot 10^6$
United Nations	$10.7 \cdot 10^6$	$300 \cdot 10^6$
Axolotl	32 books	$1 \cdot 10^6$

Books

Title	# words (approx.)
The Bible	$0.8 \cdot 10^6$
Encyclopaedia Britannica	$44 \cdot 10^6$

Aligned parallel corpora: Numbers

Corpora

Corpus	# segments (app.)	# words (app.)
JRC-Acquis	$1.0 \cdot 10^6$	$30 \cdot 10^6$
Europarl	$2.0 \cdot 10^6$	$55 \cdot 10^6$
United Nations	$10.7 \cdot 10^6$	$300 \cdot 10^6$
Axolotl	32 books	$1 \cdot 10^6$

Books

Title	# words (approx.)
The Bible	$0.8 \cdot 10^6$
Encyclopaedia Britannica	$44 \cdot 10^6$

In practice

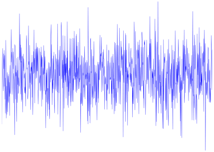
Shows real examples of the previous theory, always from freely available data/software:

- Data: `www.statmt.org/wmt13/` (Spanish–English)
- More Data: Opus, ELRC... (lots of pairs)
- Software: `SRILM`, `GIZA++` & `Moses`

Standard tools, but not exclusive

The Noisy Channel as a statistical approach to translation:

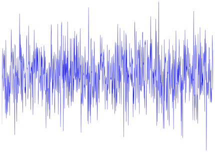
Good morning! →



SMT, basics

The Noisy Channel approach

The Noisy Channel as a statistical approach to translation:

Good m  → Bon jour!

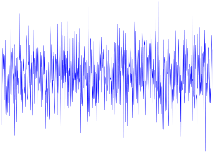
SMT, basics

The Noisy Channel approach

The Noisy Channel as a statistical approach to translation:

e: Good morning!

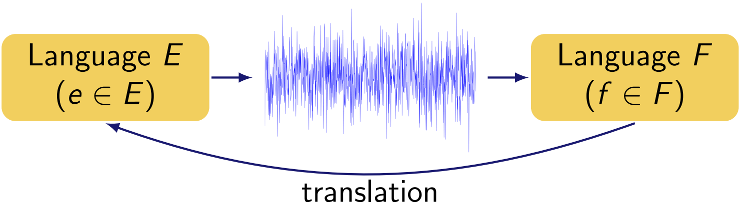
f: Bon jour!

Language E ($e \in E$) →  → Language F ($f \in F$)

translation

SMT, basics

The Noisy Channel approach

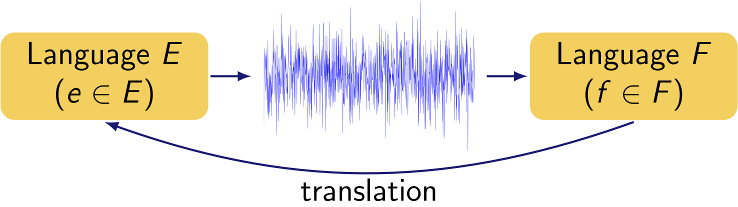


Mathematically:

$$P(e|f)$$

SMT, basics

The Noisy Channel approach



Mathematically:

$$P(e|f) = \frac{P(e)\,P(f|e)}{P(f)}$$

$$T(f) = \hat{e} = \mathrm{argmax_e}\ P(e|f) = \mathrm{argmax_e}\ P(e)\,P(f|e)$$

SMT, basics
Components

$$T(f) = \hat{e} = \mathrm{argmax_e}\ P(e)\, P(f|e)$$

Language Model

- Takes care of fluency in the target language
- Data: corpora in the target language

Translation Model

- Lexical correspondence between languages
- Data: aligned corpora in source and target languages

argmax

- Search done by the *decoder*

SMT, basics

Components

$$T(f) = \hat{e} = \mathrm{argmax_e}\ P(e)\, P(f|e)$$

Language Model

- Takes care of fluency in the target language
- Data: corpora in the target language

Translation Model

- Lexical correspondence between languages
- Data: aligned corpora in source and target languages

argmax

- Search done by the *decoder*

SMT, basics

Components

$$T(f) = \hat{e} = \mathrm{argmax}_{\mathrm{e}}\, P(e)\, P(f|e)$$

Language Model

- Takes care of fluency in the target language
- Data: corpora in the target language

Translation Model

- Lexical correspondence between languages
- Data: aligned corpora in source and target languages

argmax

- Search done by the *decoder*

Outline

Language model

$$T(f) = \hat{e} = \mathrm{argmax_e}\ P(e)\, P(f|e)$$

Estimation of how probable a sentence is.

Naïve estimation on a corpus with N sentences:

Frequentist probability of a sentence e:

$$P(e) = \frac{N_e}{N_{sentences}}$$

Problem:

- Long chains are difficult to observe in corpora.
  $\Rightarrow$ Long sentences may have zero probability!

Language model

$$T(f) = \hat{e} = \mathrm{argmax_e}\ P(e)\, P(f|e)$$

Estimation of how probable a sentence is.

Naïve estimation on a corpus with N sentences:

Frequentist probability of a sentence e:

$$P(e) = \frac{N_e}{N_{sentences}}$$

Problem:

- Long chains are difficult to observe in corpora.
  $\Rightarrow$ Long sentences may have zero probability!

Language model

$$T(f) = \hat{e} = \mathrm{argmax_e}\, P(e)\, P(f|e)$$

Estimation of how probable a sentence is.

Naïve estimation on a corpus with N sentences:

Frequentist probability of a sentence e:

$$P(e) = \frac{N_e}{N_{sentences}}$$

Problem:

- Long chains are difficult to observe in corpora.
  $\Rightarrow$ Long sentences may have zero probability!

The n-gram approach

The language model assigns a probability $P(e)$ to a sequence of words $e \Rightarrow \{w_1, \ldots, w_m\}$.

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i | w_{i-(n-1)}, \ldots, w_{i-1})$$

- The probability of a sentence is the product of the conditional probabilities of each word w_i given the previous ones.
- Independence assumption: the probability of w_i is only conditioned by the n previous words.

SMT, components

The language model $P(e)$

Example, a 4-gram model

e: `All work and no play makes Jack a dull boy`

$$P(e) = P(\texttt{All}|\phi,\phi,\phi)\ P(\texttt{work}|\phi,\phi,\texttt{All})\ P(\texttt{and}|\phi,\texttt{All},\texttt{work})$$
$$P(\texttt{no}|\texttt{All},\texttt{work},\texttt{and})\ P(\texttt{play}|\texttt{work},\texttt{and},\texttt{no})$$
$$P(\texttt{makes}|\texttt{and},\texttt{no},\texttt{play})P(\texttt{Jack}|\texttt{no},\texttt{play},\texttt{makes})$$
$$P(\texttt{a}|\texttt{play},\texttt{makes},\texttt{Jack})P(\texttt{dull}|\texttt{makes},\texttt{Jack},\texttt{a})$$
$$P(\texttt{boy}|\texttt{Jack},\texttt{a},\texttt{dull})$$

where, for each factor,

$$P(\texttt{and}|\phi,\texttt{All},\texttt{work}) = \frac{N_{(\texttt{All work and})}}{N_{(\texttt{All work})}}$$

Example, a 4-gram model

e: `All work and no play makes Jack a dull boy`

$$P(e) = P(\texttt{All}|\phi,\phi,\phi)\ P(\texttt{work}|\phi,\phi,\texttt{All})\ P(\texttt{and}|\phi,\texttt{All},\texttt{work})$$
$$P(\texttt{no}|\texttt{All},\texttt{work},\texttt{and})\ P(\texttt{play}|\texttt{work},\texttt{and},\texttt{no})$$
$$P(\texttt{makes}|\texttt{and},\texttt{no},\texttt{play}) P(\texttt{Jack}|\texttt{no},\texttt{play},\texttt{makes})$$
$$P(\texttt{a}|\texttt{play},\texttt{makes},\texttt{Jack}) P(\texttt{dull}|\texttt{makes},\texttt{Jack},\texttt{a})$$
$$P(\texttt{boy}|\texttt{Jack},\texttt{a},\texttt{dull})$$

where, for each factor,

$$P(\texttt{and}|\phi,\texttt{All},\texttt{work}) = \frac{N_{(\texttt{All work and})}}{N_{(\texttt{All work})}}$$

SMT, components

The language model $P(e)$

Example, a 4-gram model

e: `All work and no play makes Jack a dull boy`

$$P(e) = P(\texttt{All}|\phi,\phi,\phi)\ P(\texttt{work}|\phi,\phi,\texttt{All})\ P(\texttt{and}|\phi,\texttt{All},\texttt{work})$$
$$P(\texttt{no}|\texttt{All},\texttt{work},\texttt{and})\ P(\texttt{play}|\texttt{work},\texttt{and},\texttt{no})$$
$$P(\texttt{makes}|\texttt{and},\texttt{no},\texttt{play})P(\texttt{Jack}|\texttt{no},\texttt{play},\texttt{makes})$$
$$P(\texttt{a}|\texttt{play},\texttt{makes},\texttt{Jack})P(\texttt{dull}|\texttt{makes},\texttt{Jack},\texttt{a})$$
$$P(\texttt{boy}|\texttt{Jack},\texttt{a},\texttt{dull})$$

where, for each factor,

$$P(\texttt{and}|\phi,\texttt{All},\texttt{work}) = \frac{N_{(\texttt{All work and})}}{N_{(\texttt{All work})}}$$

SMT, components

The language model $P(e)$

Example, a 4-gram model

e: `All work and no play makes Jack a dull boy`

$P(e) = P(\texttt{All}|\phi,\phi,\phi)\ P(\texttt{work}|\phi,\phi,\texttt{All})\ P(\texttt{and}|\phi,\texttt{All},\texttt{work})$
$P(\texttt{no}|\texttt{All},\texttt{work},\texttt{and})\ P(\texttt{play}|\texttt{work},\texttt{and},\texttt{no})$
$P(\texttt{makes}|\texttt{and},\texttt{no},\texttt{play}) P(\texttt{Jack}|\texttt{no},\texttt{play},\texttt{makes})$
$P(\texttt{a}|\texttt{play},\texttt{makes},\texttt{Jack}) P(\texttt{dull}|\texttt{makes},\texttt{Jack},\texttt{a})$
$P(\texttt{boy}|\texttt{Jack},\texttt{a},\texttt{dull})$

where, for each factor,

$$P(\texttt{and}|\phi,\texttt{All},\texttt{work}) = \frac{N_{(\texttt{All work and})}}{N_{(\texttt{All work})}}$$

Example, a 4-gram model

e: `All work and no play makes Jack a dull boy`

$$
\begin{aligned}
P(e) = {} & P(\texttt{All}|\phi,\phi,\phi)\ P(\texttt{work}|\phi,\phi,\texttt{All})\ P(\texttt{and}|\phi,\texttt{All},\texttt{work}) \\
& P(\texttt{no}|\texttt{All},\texttt{work},\texttt{and})\ P(\texttt{play}|\texttt{work},\texttt{and},\texttt{no}) \\
& P(\texttt{makes}|\texttt{and},\texttt{no},\texttt{play}) P(\texttt{Jack}|\texttt{no},\texttt{play},\texttt{makes}) \\
& P(\texttt{a}|\texttt{play},\texttt{makes},\texttt{Jack}) P(\texttt{dull}|\texttt{makes},\texttt{Jack},\texttt{a}) \\
& P(\texttt{boy}|\texttt{Jack},\texttt{a},\texttt{dull})
\end{aligned}
$$

where, for each factor,

$$P(\texttt{and}|\phi,\texttt{All},\texttt{work}) = \frac{N_{(\texttt{All work and})}}{N_{(\texttt{All work})}}$$

SMT, components

The language model $P(e)$

Example, a 4-gram model

e: All work and no play makes Jack a dull boy

$$
\begin{aligned}
P(e) = {} & P(\texttt{All}|\phi,\phi,\phi)\ P(\texttt{work}|\phi,\phi,\texttt{All})\ P(\texttt{and}|\phi,\texttt{All},\texttt{work}) \\
& P(\texttt{no}|\texttt{All},\texttt{work},\texttt{and})\ P(\texttt{play}|\texttt{work},\texttt{and},\texttt{no}) \\
& P(\texttt{makes}|\texttt{and},\texttt{no},\texttt{play}) P(\texttt{Jack}|\texttt{no},\texttt{play},\texttt{makes}) \\
& P(\texttt{a}|\texttt{play},\texttt{makes},\texttt{Jack}) P(\texttt{dull}|\texttt{makes},\texttt{Jack},\texttt{a}) \\
& P(\texttt{boy}|\texttt{Jack},\texttt{a},\texttt{dull})
\end{aligned}
$$

where, for each factor,

$$
P(\texttt{and}|\phi,\texttt{All},\texttt{work}) = \frac{N_{(\texttt{All work and})}}{N_{(\texttt{All work})}}
$$

SMT, components

The language model $P(e)$

Example, a 4-gram model

e: `All work and no play makes Jack a dull boy`

$$
\begin{aligned}
P(e) = {} & P(\texttt{All}|\phi,\phi,\phi)\ P(\texttt{work}|\phi,\phi,\texttt{All})\ P(\texttt{and}|\phi,\texttt{All},\texttt{work}) \\
& P(\texttt{no}|\texttt{All},\texttt{work},\texttt{and})\ P(\texttt{play}|\texttt{work},\texttt{and},\texttt{no}) \\
& P(\texttt{makes}|\texttt{and},\texttt{no},\texttt{play}) P(\texttt{Jack}|\texttt{no},\texttt{play},\texttt{makes}) \\
& P(\texttt{a}|\texttt{play},\texttt{makes},\texttt{Jack}) P(\texttt{dull}|\texttt{makes},\texttt{Jack},\texttt{a}) \\
& P(\texttt{boy}|\texttt{Jack},\texttt{a},\texttt{dull})
\end{aligned}
$$

where, for each factor,

$$
P(\texttt{and}|\phi,\texttt{All},\texttt{work}) = \frac{N_{(\texttt{All work and})}}{N_{(\texttt{All work})}}
$$

Example, a 4-gram model

e: `All work and no play makes Jack a dull boy`

$$
\begin{aligned}
P(e) = {} & P(\texttt{All}|\phi,\phi,\phi)\ P(\texttt{work}|\phi,\phi,\texttt{All})\ P(\texttt{and}|\phi,\texttt{All},\texttt{work}) \\
& P(\texttt{no}|\texttt{All},\texttt{work},\texttt{and})\ P(\texttt{play}|\texttt{work},\texttt{and},\texttt{no}) \\
& P(\texttt{makes}|\texttt{and},\texttt{no},\texttt{play}) P(\texttt{Jack}|\texttt{no},\texttt{play},\texttt{makes}) \\
& P(\texttt{a}|\texttt{play},\texttt{makes},\texttt{Jack}) P(\texttt{dull}|\texttt{makes},\texttt{Jack},\texttt{a}) \\
& P(\texttt{boy}|\texttt{Jack},\texttt{a},\texttt{dull})
\end{aligned}
$$

where, for each factor,

$$P(\text{and}|\phi, \texttt{All}, \texttt{work}) = \frac{N_{(\texttt{All work and})}}{N_{(\texttt{All work})}}$$

SMT, components

The language model $P(e)$

Example, a 4-gram model

e: `All work and no play makes Jack a dull boy`

$$
\begin{aligned}
P(e) = {} & P(\texttt{All}|\phi,\phi,\phi)\ P(\texttt{work}|\phi,\phi,\texttt{All})\ P(\texttt{and}|\phi,\texttt{All},\texttt{work}) \\
& P(\texttt{no}|\texttt{All},\texttt{work},\texttt{and})\ P(\texttt{play}|\texttt{work},\texttt{and},\texttt{no}) \\
& P(\texttt{makes}|\texttt{and},\texttt{no},\texttt{play}) P(\texttt{Jack}|\texttt{no},\texttt{play},\texttt{makes}) \\
& P(\texttt{a}|\texttt{play},\texttt{makes},\texttt{Jack}) P(\texttt{dull}|\texttt{makes},\texttt{Jack},\texttt{a}) \\
& P(\texttt{boy}|\texttt{Jack},\texttt{a},\texttt{dull})
\end{aligned}
$$

where, for each factor,

$$P(\texttt{and}|\phi,\texttt{All},\texttt{work}) = \frac{N_{(\texttt{All work and})}}{N_{(\texttt{All work})}}$$

Example, a 4-gram model

e: `All work and no play makes Jack a dull boy`

$$
\begin{aligned}
P(e) = {} & P(\texttt{All}|\phi,\phi,\phi)\ P(\texttt{work}|\phi,\phi,\texttt{All})\ P(\texttt{and}|\phi,\texttt{All},\texttt{work}) \\
& P(\texttt{no}|\texttt{All},\texttt{work},\texttt{and})\ P(\texttt{play}|\texttt{work},\texttt{and},\texttt{no}) \\
& P(\texttt{makes}|\texttt{and},\texttt{no},\texttt{play}) P(\texttt{Jack}|\texttt{no},\texttt{play},\texttt{makes}) \\
& P(\texttt{a}|\texttt{play},\texttt{makes},\texttt{Jack}) P(\texttt{dull}|\texttt{makes},\texttt{Jack},\texttt{a}) \\
& P(\texttt{boy}|\texttt{Jack},\texttt{a},\texttt{dull})
\end{aligned}
$$

where, for each factor,

$$P(\texttt{and}|\phi,\texttt{All},\texttt{work}) = \frac{N_{(\texttt{All work and})}}{N_{(\texttt{All work})}}$$

Example, a 4-gram model

e: All work and no play makes Jack a dull boy

$$
\begin{aligned}
P(e) = {} & P(\texttt{All}|\phi,\phi,\phi)\ P(\texttt{work}|\phi,\phi,\texttt{All})\ P(\texttt{and}|\phi,\texttt{All},\texttt{work}) \\
& P(\texttt{no}|\texttt{All},\texttt{work},\texttt{and})\ P(\texttt{play}|\texttt{work},\texttt{and},\texttt{no}) \\
& P(\texttt{makes}|\texttt{and},\texttt{no},\texttt{play}) P(\texttt{Jack}|\texttt{no},\texttt{play},\texttt{makes}) \\
& P(\texttt{a}|\texttt{play},\texttt{makes},\texttt{Jack}) P(\texttt{dull}|\texttt{makes},\texttt{Jack},\texttt{a}) \\
& P(\texttt{boy}|\texttt{Jack},\texttt{a},\texttt{dull})
\end{aligned}
$$

where, for each factor,

$$P(\texttt{and}|\phi,\texttt{All},\texttt{work}) = \frac{N_{(\texttt{All work and})}}{N_{(\texttt{All work})}}$$

SMT, components

The language model $P(e)$

Example, a 4-gram model

e: `All work and no play makes Jack a dull boy`

$$
\begin{aligned}
P(e) = {} & P(\texttt{All}|\phi,\phi,\phi)\ P(\texttt{work}|\phi,\phi,\texttt{All})\ P(\texttt{and}|\phi,\texttt{All},\texttt{work}) \\
& P(\texttt{no}|\texttt{All},\texttt{work},\texttt{and})\ P(\texttt{play}|\texttt{work},\texttt{and},\texttt{no}) \\
& P(\texttt{makes}|\texttt{and},\texttt{no},\texttt{play})P(\texttt{Jack}|\texttt{no},\texttt{play},\texttt{makes}) \\
& P(\texttt{a}|\texttt{play},\texttt{makes},\texttt{Jack})P(\texttt{dull}|\texttt{makes},\texttt{Jack},\texttt{a}) \\
& P(\texttt{boy}|\texttt{Jack},\texttt{a},\texttt{dull})
\end{aligned}
$$

where, for each factor,

$$
P(\texttt{and}|\phi,\texttt{All},\texttt{work}) = \frac{N_{(\texttt{All work and})}}{N_{(\texttt{All work})}}
$$

Main problems and criticisims:

- Long-range dependencies are lost.
- Still, some *n*-grams can be not observed in the corpus.

Solution

Smoothing techniques:

- Linear interpolation.

$$P(\texttt{and}|\texttt{All},\texttt{work}) = \frac{N_{(\texttt{All},\texttt{work},\texttt{and})}}{N_{(\texttt{All},\texttt{work})}} + \lambda_2 \frac{N_{(\texttt{work},\texttt{and})}}{N_{(\texttt{work})}} + \lambda_1 \frac{N_{(\texttt{and})}}{N_{words}} + \lambda_0$$

SMT, components

The language model $P(e)$

Main problems and criticisims:

- Long-range dependencies are lost.
- Still, some *n*-grams can be not observed in the corpus.

Solution

Smoothing techniques:

- Linear interpolation.
- Back-off models.

$$P(\texttt{and}|\texttt{All},\texttt{work}) = \frac{N_{(\texttt{All},\texttt{work},\texttt{and})}}{N_{(\texttt{All},\texttt{work})}} + \lambda_2 \frac{N_{(\texttt{work},\texttt{and})}}{N_{(\texttt{work})}} + \lambda_1 \frac{N_{(\texttt{and})}}{N_{words}} + \lambda_0$$

SMT, components

The language model $P(e)$

Main problems and criticisims:

- Long-range dependencies are lost.
- Still, some *n*-grams can be not observed in the corpus.

Solution

Smoothing techniques:

- Linear interpolation.

$$P(\texttt{and}|\texttt{All}, \texttt{work}) = \frac{N_{(\texttt{All},\texttt{work},\texttt{and})}}{N_{(\texttt{All},\texttt{work})}} + \lambda_2 \frac{N_{(\texttt{work},\texttt{and})}}{N_{(\texttt{work})}} + \lambda_1 \frac{N_{(\texttt{and})}}{N_{words}} + \lambda_0$$

Main problems and criticisims:

- Long-range dependencies are lost.
- Still, some *n*-grams can be not observed in the corpus.

Solution

Smoothing techniques:

- Linear interpolation.

$$P(\texttt{and}|\texttt{All},\texttt{work}) = \lambda_3 \frac{N_{(\texttt{All},\texttt{work},\texttt{and})}}{N_{(\texttt{All},\texttt{work})}} + \lambda_2 \frac{N_{(\texttt{work},\texttt{and})}}{N_{(\texttt{work})}} + \lambda_1 \frac{N_{(\texttt{and})}}{N_{words}} + \lambda_0$$

In practice,

```
cluster:/home/quest/corpus/lm> ls -lkh

-rw-r--r-- 1  emt ia   507M   mar  3 15:28 europarl.lm
-rw-r--r-- 1  emt ia    50M   mar  3 15:29 nc.lm
-rw-r--r-- 1  emt ia   3,1G   mar  3 15:33 un.lm

cluster:/home/quest/corpus/lm> wc -l

  15,181,883   europarl.lm
   1,735,721   nc.lm
  82,504,380   un.lm
```

SMT, components

The language model $P(e)$

```
cluster:/home/quest/corpus/lm> more nc.lm


\data\
ngram 1=655770
ngram 2=11425501
ngram 3=10824125
ngram 4=13037011
ngram 5=12127575

\1-grams:
-3.142546 ! -1.415594
-1.978775 " -0.9078496
-4.266428 # -0.2729652
-3.806078 $ -0.3918373
-3.199419 % -1.139753
-3.613416 & -0.6046973
-2.712332 '-0.6271471
-2.268107 ( -0.6895114
```

SMT, components

The language model $P(e)$

```
\2-grams:
 -1.08232 concierto ,
-1.093977 concierto . -0.2378127
-1.747908 concierto ad
-1.748422 concierto cobraria
-0.8927398 concierto de
-1.744176 concierto europeo
-1.740879 concierto internacional
-1.635606 concierto para
-1.744787 concierto regional

...

\5-grams:
-0.8890668 no son los unicos culpables
-1.396196 no son los unicos problemas
-0.7550655 no son los unicos que
-1.240193 no son los unicos responsables
```

Language model: keep in mind

- Statistical LMs estimate the probability of a sentence from its n-gram frequency counts in a monolingual corpus.
- Within an SMT system, it contributes to select fluent sentences in the target language.
- Smoothing techniques are used so that not frequent translations are not discarded beforehand.

SMT, components

The translation model $P(f|e)$

Translation model

$$T(f) = \hat{e} = \mathrm{argmax}_e \; P(e) \, P(f|e)$$

Estimation of the lexical correspondence between languages.

How can be $P(f|e)$ characterised?

SMT, components

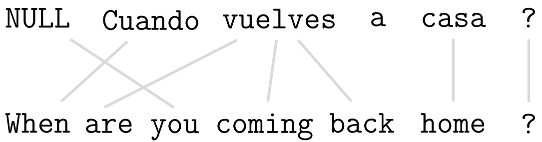
The translation model $P(f|e)$

Translation model

$$T(f) = \hat{e} = \mathrm{argmax}_{\mathrm{e}}\ P(e)\, P(f|e)$$

Estimation of the lexical correspondence between languages.

How can be $P(f|e)$ characterised?



SMT, components

The translation model $P(f|e)$

Translation model

$$T(f) = \hat{e} = \mathrm{argmax_e}\ P(e)\,P(f|e)$$

Estimation of the lexical correspondence between languages.

How can be $P(f|e)$ characterised?

SMT, components

The translation model $P(f|e)$

One should at least model for *each word* in the source language:

- Its translation,
- the number of necessary words in the target language,
- the position of the translation within the sentence,
- and, besides, the number of words that need to be generated from scratch.

Word-based models: the IBM models

They characterise $P(f|e)$ with 4 parameters: t, n, d and p_1.

- Lexical probability t
  $t(\texttt{Cuando}|\texttt{When})$: the prob. that `Cuando` translates into `When`.

- Fertility n
  $n(3|\texttt{vuelves})$: the prob. that `vuelves` generates 3 words.

SMT, components

The translation model $P(f|e)$

Word-based models: the IBM models

They characterise $P(f|e)$ with 4 parameters: t, n, d and p_1.

- Distortion d
  $d(j|i, m, n)$: the prob. that the word in the j position generates a word in the i position. m and n are the length of the source and target sentences.
- Probability p_1
  $p(\texttt{you}|\texttt{NULL})$: the prob. that the spurious word `you` is generated (from `NULL`).

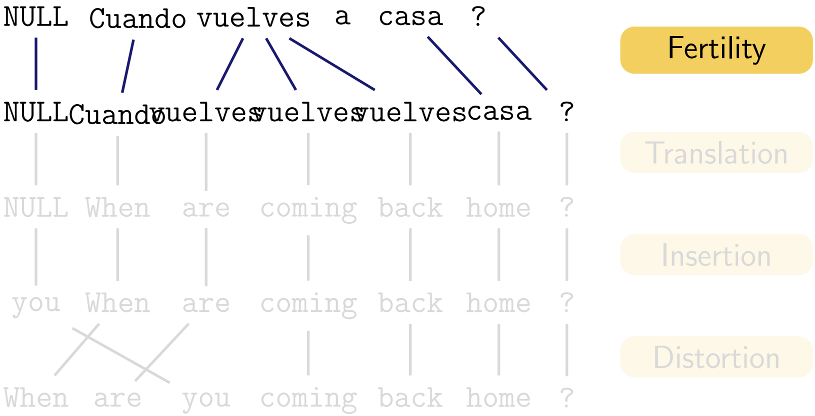
Back to the example:

`NULL Cuando vuelves a casa ?`

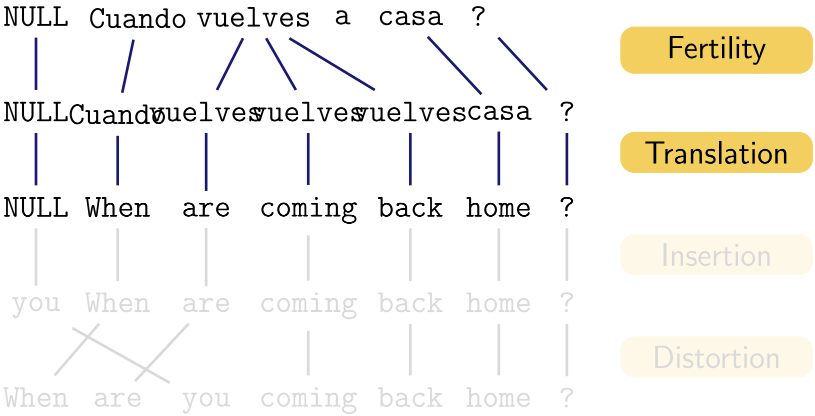
SMT, components

The translation model $P(f|e)$

Back to the example:



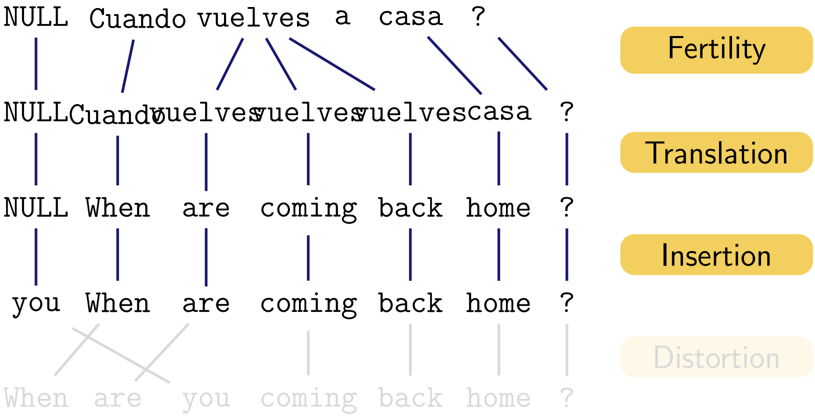
Back to the example:



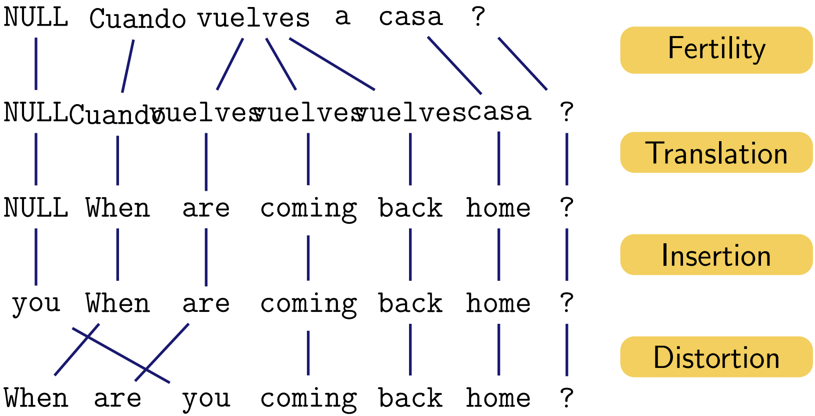
SMT, components

The translation model $P(f|e)$

Back to the example:



Back to the example:



SMT, components

The translation model $P(f|e)$

Word-based models: the IBM models

How can t, n, d and p_1 be estimated?

- Statistical model $\Rightarrow$ counts in a (huge) corpus!

But...

- Corpora are aligned at sentence level, not at word level.

Alternatives

- Pay someone to align 2 milion sentences word by word.
- Estimate word alignments together with the parameters.

Word-based models: the IBM models

How can t, n, d and p_1 be estimated?

- Statistical model $\Rightarrow$ counts in a (huge) corpus!

But...

- Corpora are aligned at sentence level, not at word level.

Alternatives

- Pay someone to align 2 milion sentences word by word.
- Estimate word alignments together with the parameters.

Word-based models: the IBM models

How can t, n, d and p_1 be estimated?

- Statistical model $\Rightarrow$ counts in a (huge) corpus!

But...

- Corpora are aligned at sentence level, not at word level.

Alternatives

- Pay someone to align 2 milion sentences word by word.
- Estimate word alignments together with the parameters.

SMT, components

The translation model $P(f|e)$

Expectation-Maximisation algorithm

Parameter initialisation

↓

Alignment probability calculation

SMT, components

The translation model $P(f|e)$

Expectation-Maximisation algorithm

Parameter initialisation

↓

Alignment probability calculation

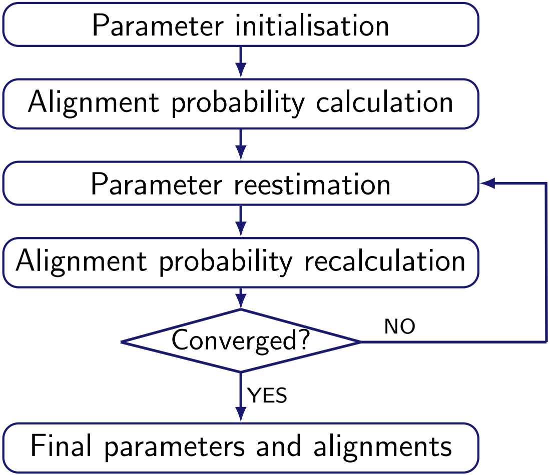
↓

Parameter reestimation

↓

Alignment probability recalculation

Expectation-Maximisation algorithm

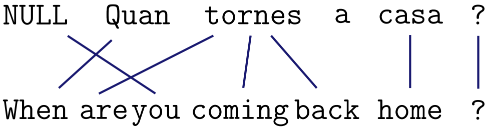


Alignment's asymmetry

The definitions in IBM models make the alignments asymmetric

- each target word corresponds to only one source word, but the opposite is not true due to the definition of fertility.

Catalan to English



English to Catalan

SMT, components

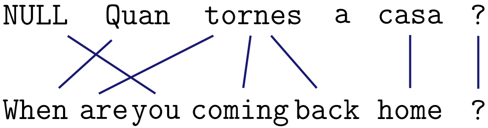
The translation model $P(f|e)$

Alignment's asymmetry

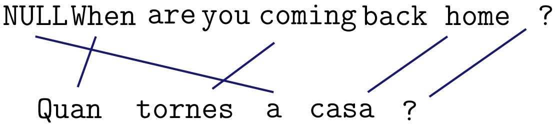
The definitions in IBM models make the alignments asymmetric

- each target word corresponds to only one source word, but the opposite is not true due to the definition of fertility.

Catalan to English



English to Catalan



Visually:

Catalan to English

Visually:

English to Catalan

Alignment symmetrisation

- Intersection: high-confidence, high precision.

Catalan to English $\bigcap$ English to Catalan

Alignment symmetrisation

- Union: lower confidence, high recall.

Catalan to English $\bigcup$ English to Catalan

In practice,

```
cluster:/home/moses/giza.en-es> zmore en-es.A3.final.gz
```

```
# Sentence pair (1) source length 5 target length 4 alignment score: 0.00015062
resumption of the session
NULL ({ }) reanudacion ({ 1 }) del ({ 2 3 }) periodo ({ }) de ({ }) sesiones ({ 4 })

# Sentence pair (2) source length 33 target length 40 alignment score: 3.3682e-61
i declare resumed the session of the european parliament adjourned on friday 17
december 1999 , and i would like once again to wish you a happy new year in the
hope that you enjoyed a pleasant festive period .
NULL ({ 31 }) declaro ({ 1 }) reanudado ({ 2 3 }) el ({ 4 }) periodo ({ }) de ({ })
sesiones ({ 5 }) del ({ 6 7 }) parlamento ({ 9 }) europeo ({ 8 }) , ({ })
interrumpido ({ 10 }) el ({ }) viernes ({ 12 14 }) 17 ({ 11 13 }) de ({ }) diciembre
({ 15 }) pasado ({ }) , ({ 16 }) y ({ 17 }) reitero ({ 21 }) a ({ 23 }) sus ({ 30 })
senorias ({ }) mi ({ 18 }) deseo ({ 24 }) de ({ }) que ({ 33 }) hayan ({ 25 34 35 })
 tenido ({ }) unas ({ 19 20 }) buenas ({ 26 36 }) vacaciones ({ 22 27 28 29 32 37 38
 39 }) . ({ 40 })
```

SMT, components

The translation model $P(f|e)$

In practice,

```
cluster:/home/moses/giza.es-en> zmore es-en.A3.final.gz
```

```
# Sentence pair (1) source length 4 target length 5 alignment score: 1.08865e-07
reanudacion del periodo de sesiones
NULL ({ 4 }) resumption ({ 1 }) of ({ 2 }) the ({ }) session ({ 3 5 })

# Sentence pair (2) source length 40 target length 33 alignment score: 1.88268e-50
declaro reanudado el periodo de sesiones del parlamento europeo , interrumpido el
viernes 17 de diciembre pasado , y reitero a sus senorias mi deseo de que hayan
tenido unas buenas vacaciones .
NULL ({ 5 10 }) i ({ }) declare ({ 1 }) resumed ({ 2 }) the ({ 3 }) session ({ 4 6 })
of ({ 7 }) the ({ }) european ({ 9 }) parliament ({ 8 12 }) adjourned ({ 11 }) on
({ 15 }) friday ({ 13 }) 17 ({ 14 }) december ({ 16 17 }) 1999 ({ }) , ({ 18 }) and
({ 19 }) i ({ }) would ({ }) like ({ }) once ({ }) again ({ }) to ({ 21 }) wish ({ })
you ({ }) a ({ }) happy ({ }) new ({ }) year ({ }) in ({ 26 }) the ({ }) hope ({ }
) that ({ 27 }) you ({ }) enjoyed ({ 20 }) a ({ }) pleasant ({ 22 23 24 25 28 29 })
festive ({ 30 31 32 }) period ({ }) . ({ 33 })
```

SMT, components

The translation model $P(f|e)$

```
cluster:/home/moses/model> more aligned.grow-diag-final

0-0 1-1 1-2 2-3 4-3

0-0 0-1 1-1 1-2 2-3 3-4 5-4 6-5 6-6 8-7 7-8 11-8 10-9 13-10 14-10 12-11
13-12 12-13 15-14 17-15 18-16 23-17 19-20 20-22 24-23 21-29 26-32 27-33
27-34 30-35 28-36 31-36 29-37 30-37 31-37 31-38 32-39
```

SMT, components

The translation model $P(f|e)$

```
cluster:/home/moses/model> more lex.e2f

tuneles tunnels 0.7500000
tuneles transit 0.2000000
estructuralmente weak 1.0000000
estructuralmente structurally 0.5000000
destruido had 0.0454545
para tunnels 0.2500000
sean transit 0.2000000
transito transit 0.6000000
...

cluster:/home/moses/model> more lex.f2e

tunnels tuneles 0.7500000
transit tuneles 0.2500000
weak estructuralmente 0.5000000
structurally estructuralmente 0.5000000
...
```

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: ϕ

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the book`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the book new.`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the book new.` $\sim$

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: ϕ

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: `David`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: `David reads`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: `David reads the`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: `David reads the book`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: `David reads the book of`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: `David reads the book of new.`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: `David reads the book of new.` ✗

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: `David reads the book of new.` ✗

e: ϕ

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: `David reads the book of new.` ✗

e: `David`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: `David reads the book of new.` ✗

e: `David reads`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: `David reads the book of new.` ✗

e: `David reads the`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: `David reads the book of new.` ✗

e: `David reads the book`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: `David reads the book of new.` ✗

e: `David reads the book again.`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

f: `En David llegeix el llibre de nou.`

e: `David reads the book of new.` ✗
e: `David reads the book again.` ✓

From Word-based to Phrase-based models

f: `En David llegeix el llibre` $\boxed{\texttt{nou}}$.

e: `David reads the` $\boxed{\texttt{new}}$ `book.` ✓

f: `En David llegeix el llibre` $\boxed{\texttt{de nou}}$.

e: `David reads the book of new.` ✗

e: `David reads the book` $\boxed{\texttt{again}}$. ✓

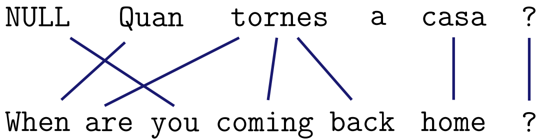
- Some sequences of words usually translate together.
- Approach: take sequences (phrases) as translation units.

What can be achieved with phrase-based models
(as compared to word-based models)

- Allow to translate from several to several words and not only from one to several.
- Some local and short range context is used.
- Idioms can be catched.

SMT, components

The translation model $P(f|e)$

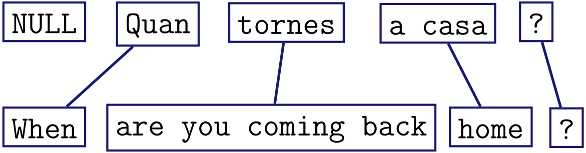


With the new translation units, $P(f|e)$ can be obtained following the same strategy as for word-based models with few modifications:

1. Segment source sentence into phrases.
2. Translate each phrase into the target language.
3. Reorder the output.

SMT, components

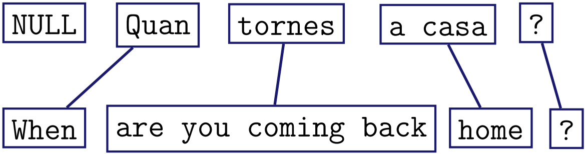
The translation model $P(f|e)$



With the new translation units, $P(f|e)$ can be obtained following the same strategy as for word-based models with few modifications:

1. Segment source sentence into phrases.
2. Translate each phrase into the target language.
3. Reorder the output.

SMT, components
The translation model $P(f|e)$

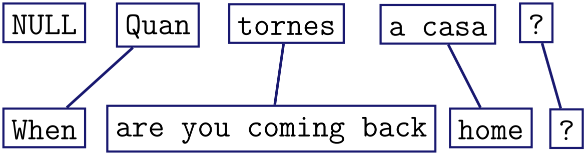


With the new translation units, $P(f|e)$ can be obtained following the same strategy as for word-based models with few modifications:

1. Segment source sentence into phrases.
2. Translate each phrase into the target language.
3. Reorder the output.

SMT, components

The translation model $P(f|e)$



But...

- Alignments need to be done at phrase level

Options

- Calculate phrase-to-phrase alignments $\Rightarrow$ hard!
- Obtain phrase alignments from word alignments $\Rightarrow$ how?

SMT, components

The translation model $P(f|e)$

Questions to answer:

- How do we obtain phrase alignments from word alignments?
- And, by the way, what's exactly a phrase?!

A **phrase is** a sequence of words consistent with word alignment.
That is, no word is aligned to a word outside the phrase.
But a phrase **is not** necessarily a linguistic element.

[1] We do not use the term phrase here in its linguistic sense: a phrase can be any sequence of words, even if they are not a linguistic constituent.

SMT, components

The translation model $P(f|e)$

Questions to answer:

- How do we obtain phrase alignments from word alignments?
- And, by the way, what's exactly a phrase?!

A **phrase is** a sequence of words consistent with word alignment.
That is, no word is aligned to a word outside the phrase.
But a phrase **is not** necessarily a linguistic element.

[1] We do not use the term phrase here in its linguistic sense: a phrase can be any sequence of words, even if they are not a linguistic constituent.

Questions to answer:

- How do we obtain phrase alignments from word alignments?
- And, by the way, what's exactly a phrase?!

A **phrase is** a sequence of words consistent with word alignment.
That is, no word is aligned to a word outside the phrase.
But a phrase **is not** necessarily a linguistic element.

[1] We do not use the term phrase here in its linguistic sense: a phrase can be any sequence of words, even if they are not a linguistic constituent.

SMT, components

The translation model $P(f|e)$

Questions to answer:

- How do we obtain phrase alignments from word alignments?
- And, by the way, what's exactly a phrase?!

A **phrase is** a sequence of words consistent with word alignment.
That is, no word is aligned to a word outside the phrase.
But a phrase **is not** necessarily a linguistic element.[1]

[1] We do not use the term phrase here in its linguistic sense: a phrase can be any sequence of words, even if they are not a linguistic constituent.

SMT, components

The translation model $P(f|e)$

Phrase extraction through an example:

(Quan tornes, When are you coming back)

SMT, components

The translation model $P(f|e)$

Phrase extraction through an example:

~~(Quan tornes, When are you coming back)~~

SMT, components

The translation model $P(f|e)$

Phrase extraction through an example:

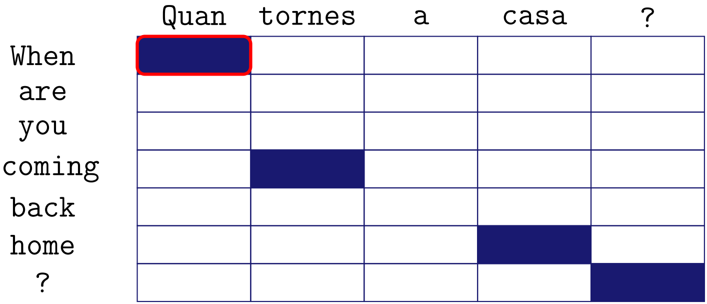
~~(Quan tornes, When are you coming back)~~

(Quan tornes tu, When are you coming back)

SMT, components

The translation model $P(f|e)$

Intersection

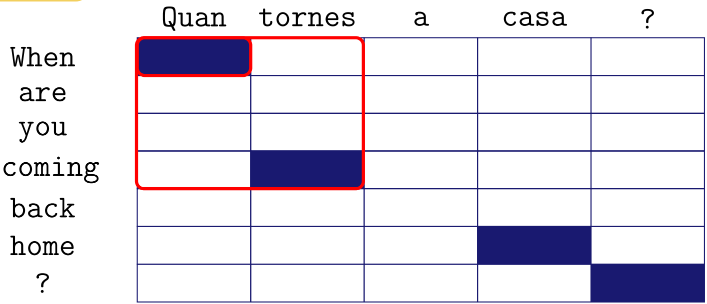


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

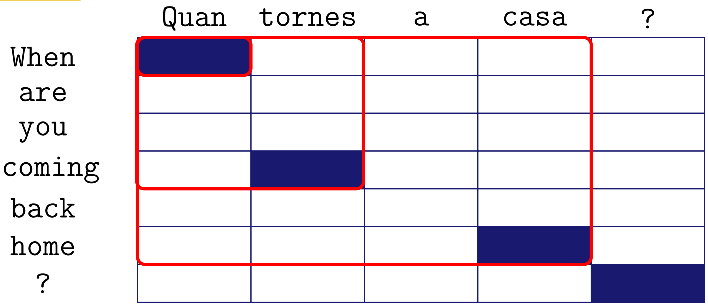


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

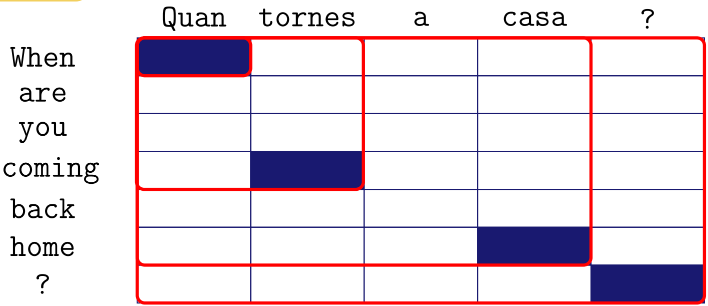


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

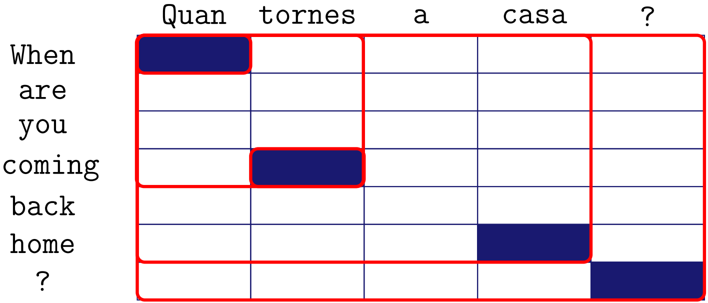


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

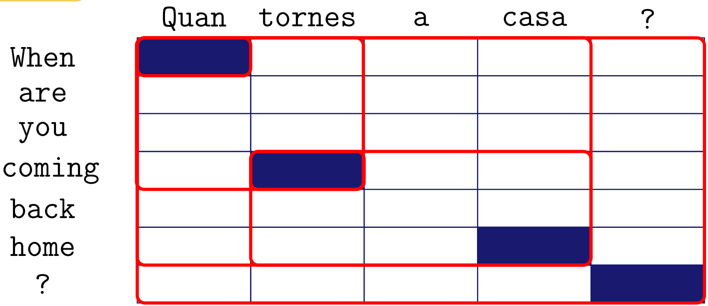


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

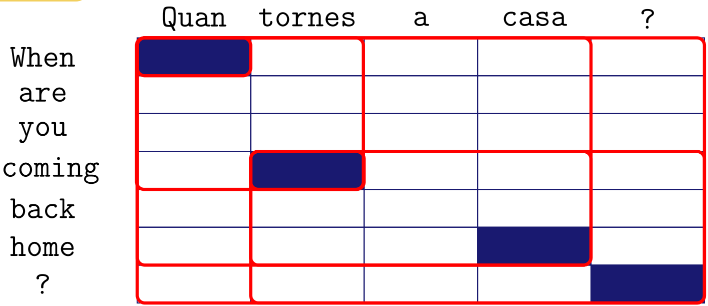


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

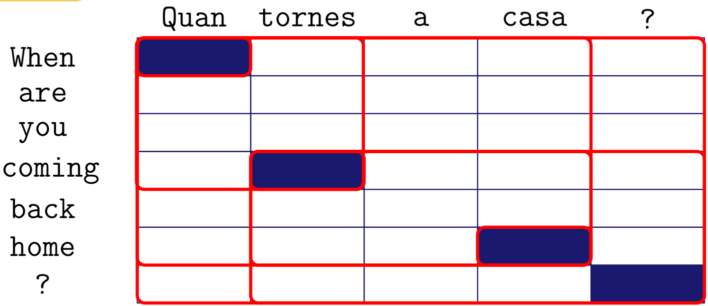
Intersection



(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components
The translation model $P(f|e)$

Intersection

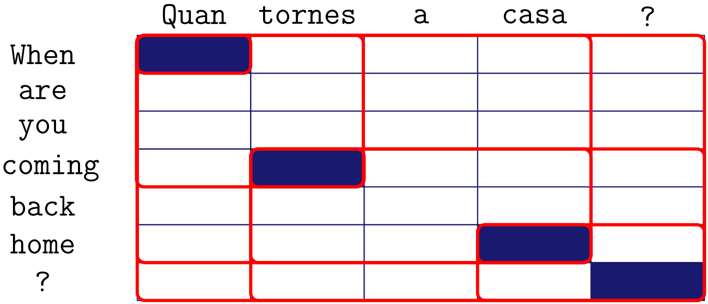


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

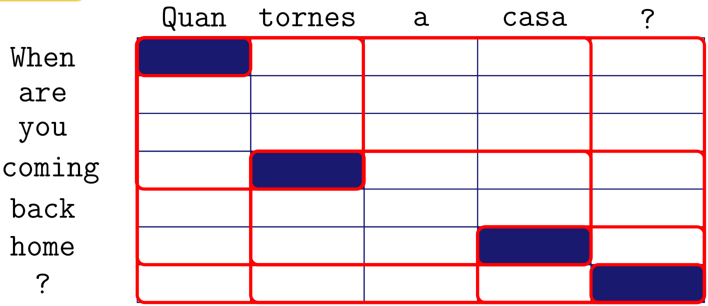


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

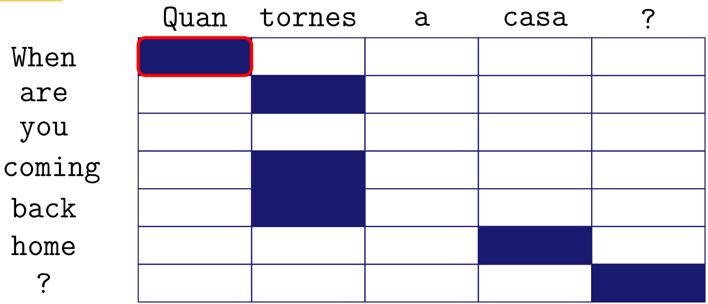


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Union

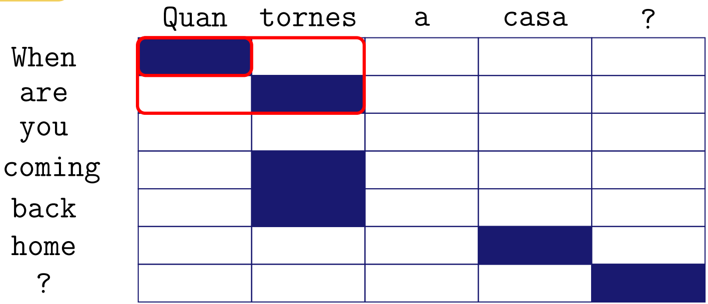


(Quan, When) (Quan tornes, When are) (Quan tornes, When are you coming) (Quan tornes, When are you coming back) (Quan tornes a casa, When are you coming back home) ... (tornes a casa ?, are you coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 21 phrases

SMT, components

The translation model $P(f|e)$

Union

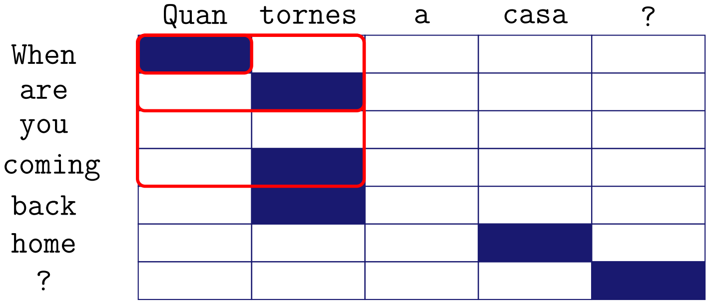


(Quan, When) (Quan tornes, When are) (Quan tornes, When are you coming) (Quan tornes, When are you coming back) (Quan tornes a casa, When are you coming back home) ... (tornes a casa ?, are you coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 21 phrases

SMT, components

The translation model $P(f|e)$

Union

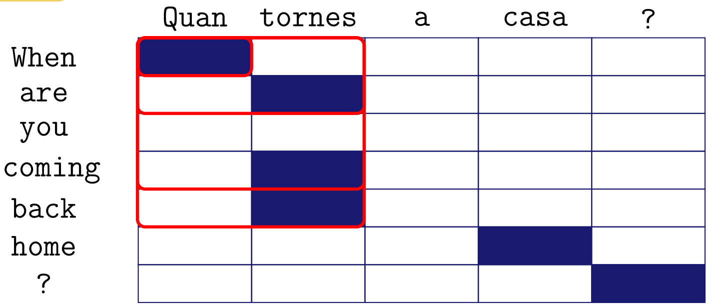


(Quan, When) (Quan tornes, When are) (Quan tornes, When are you coming) (Quan tornes, When are you coming back) (Quan tornes a casa, When are you coming back home) ... (tornes a casa ?, are you coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 21 phrases

SMT, components

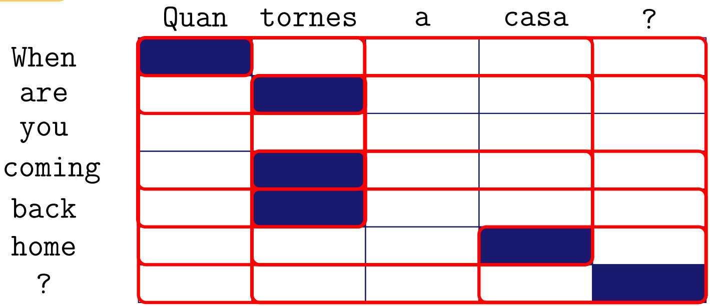
The translation model $P(f|e)$

Union



(Quan, When) (Quan tornes, When are) (Quan tornes, When are you coming) (Quan tornes, When are you coming back) (Quan tornes a casa, When are you coming back home) ... (tornes a casa ?, are you coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 21 phrases

Union



(Quan, When) (Quan tornes, When are) (Quan tornes, When are you coming) (Quan tornes, When are you coming back) (Quan tornes a casa, When are you coming back home) ... (tornes a casa ?, are you coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 21 phrases

Phrase extraction

- The number of extracted phrases depends on the symmetrisation method.
  - Intersection: few precise phrases.
  - Union: lots of (less?) precise phrases.
- Usually, neither intersection nor union are used, but something in between.
  - Start from the intersection and add points belonging to the union according to heuristics.

Phrase extraction

- For each phrase-pair (f_i, e_i), $P(f_i|e_i)$ is estimated by frequency counts in the parallel corpus.
- The set of possible phrase-pairs conforms the set of translation options.
- The set of phrase-pairs together with their probabilities conform the translation table.

In practice,

```
cluster:/home/moses/model> zmore extract.gz
```

```
reanudacion ||| resumption ||| 0-0
reanudacion del ||| resumption of the ||| 0-0 1-1 1-2
reanudacion del periodo de sesiones ||| resumption of the session ||| 0-0 1-1 1-2 2-3 4-3
```

```
cluster:/home/moses/model> zmore extract.inv.gz
```

```
resumption ||| reanudacion ||| 0-0
resumption of the ||| reanudacion del ||| 0-0 1-1 2-1
resumption of the session ||| reanudacion del periodo de sesiones ||| 0-0 1-1 2-1 3-2 3-4
```

```
cluster:/home/moses/model> zmore extract.o.gz
```

```
reanudacion ||| resumption ||| mono mono
reanudacion del ||| resumption of the ||| mono mono
reanudacion del periodo de sesiones ||| resumption of the session ||| mono mono
```

SMT, components

The translation model $P(f|e)$

```
cluster:/home/moses/model> zmore phrase-table.gz
```

```
be consistent ||| coherentes ||| 0.0384615 0.146893 0.0833333 0.0116792 2.718 ||| 1-0 ||| 26 12
be consistent ||| sean coherentes ||| 0.2 0.00022714 0.0833333 0.0916808 2.718 ||| 0-0 1-1 ||| 5 12
be consistent ||| sean consistentes ||| 0.5 0.000104834 0.0833333 0.0785835 2.718 ||| 0-0 1-1 ||| 2 12
be consistent ||| ser coherente ||| 0.5 0.0204044 0.166667 0.569957 2.718 ||| 0-0 1-1 ||| 4 12
be consistent ||| ser consecuente ||| 1 0.000340072 0.0833333 0.759942 2.718 ||| 0-0 1-1 ||| 1 12
be consistent ||| ser consistente ||| 1 0.00850183 0.5 0.633285 2.718 ||| 0-0 1-1 ||| 6 12
consistent when ||| coherente cuando se ||| 1 0.00783857 1 0.329794 2.718 ||| 0-0 1-1 1-2 ||| 1 1
consistent ||| adecuado ||| 0.00512821 0.0112994 0.00671141 0.009009 2.718 ||| 0-0 ||| 195 149
consistent ||| coherencia ||| 0.137931 0.0282486 0.0268456 0.0847458 2.718 ||| 0-0 ||| 29 149
consistent ||| constante ||| 0.0333333 0.0112994 0.0134228 0.0307692 2.718 ||| 0-0 ||| 60 149
consistent ||| constantes ||| 0.0625 0.0056497 0.00671141 0.047619 2.718 ||| 0-0 ||| 16 149
...
```

Translation model: keep in mind

- Statistical TMs estimate the probability of a translation from a parallel aligned corpus.
- Its quality depends on the quality of the obtained word (phrase) alignments.
- Within an SMT system, it contributes to select semantically adequate sentences in the target language.

¡Gracias! ¿Preguntas?

Statistical Machine Translation: Main Components

Cristina España i Bonet
DFKI GmbH

1er. Congreso Internacional de
Procesamiento de Lenguaje Natural para Lenguas Indígenas

Morelia, Michoacán, México
5th November, 2020

Decoder

$$T(f) = \hat{e} = \mathrm{argmax}_{\mathrm{e}}\; P(e)\, P(f|e)$$

Responsible for the search in the space of possible translations.

Given a model (LM+TM+...), the decoder constructs the possible translations and looks for the most probable one.

In our context, one can find:

- Greedy decoders. Initial hypothesis (word by word translation) refined iteratively using hill-climbing heuristics.
- Beam search decoders.

Decoder

$$T(f) = \hat{e} = \mathrm{argmax_e}\; P(e)\, P(f|e)$$

Responsible for the search in the space of possible translations.

Given a model (LM+TM+...), the decoder constructs the possible translations and looks for the most probable one.

In our context, one can find:

- Greedy decoders. Initial hypothesis (word by word translation) refined iteratively using hill-climbing heuristics.
- Beam search decoders.

Decoder

$$T(f) = \hat{e} = \mathrm{argmax}_\mathrm{e}\; P(e)\, P(f|e)$$

Responsible for the search in the space of possible translations.

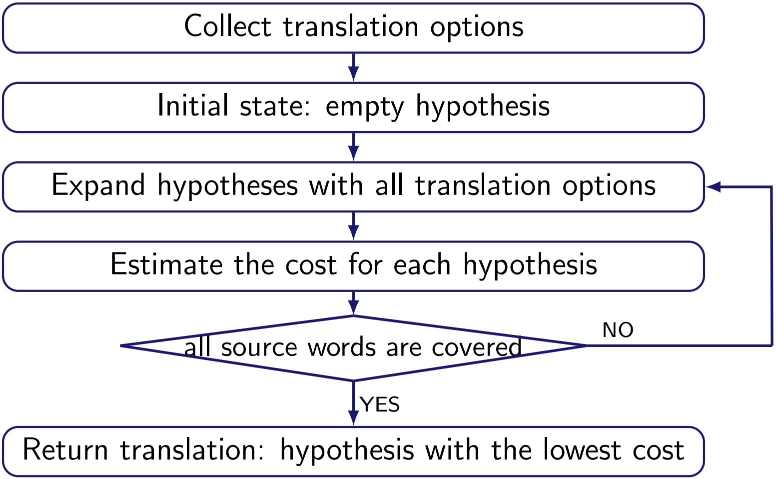
Given a model (LM+TM+...), the decoder constructs the possible translations and looks for the most probable one.

In our context, one can find:

- Greedy decoders. Initial hypothesis (word by word translation) refined iteratively using hill-climbing heuristics.
- Beam search decoders. Let's see..

SMT, components
Decoding

Core algorithm



Example: `Quan tornes a casa`

- Translation options:

  (Quan, When)
  (Quan_tornes, When_are_you_coming_back)
  (Quan_tornes_a_casa, When_are_you_coming_back_home)
  (tornes, come_back)
  (tornes_a_casa, come_back_home)
  (a_casa, home)

Example: `Quan tornes a casa`

- Translation options:

  (Quan, When)
  (Quan_tornes, When_are_you_coming_back)
  (Quan_tornes_a_casa, When_are_you_coming_back_home)
  (tornes, come_back)
  (tornes_a_casa, come_back_home)
  (a_casa, home)

- Notation for hypotheses in construction:

  Constructed sentence so far: come_back
  Source words already translated: - x - -

Example: `Quan tornes a casa`

- Translation options:

  (`Quan`, `When`)
  (`Quan_tornes`, `When_are_you_coming_back`)
  (`Quan_tornes_a_casa`, `When_are_you_coming_back_home`)
  (`tornes`, `come_back`)
  (`tornes_a_casa`, `come_back_home`)
  (`a_casa`, `home`)

- Notation for hypotheses in construction:

  Constructed sentence so far: `come_back`
  Source words already translated: - x - -

Example: `Quan tornes a casa`

- Translation options:

  (Quan, When)
  (Quan_tornes, When_are_you_coming_back)
  (Quan_tornes_a_casa, When_are_you_coming_back_home)
  (tornes, come_back)
  (tornes_a_casa, come_back_home)
  (a_casa, home)

- Initial hypothesis

  Constructed sentence so far: ϕ
  Source words already translated: - - - -

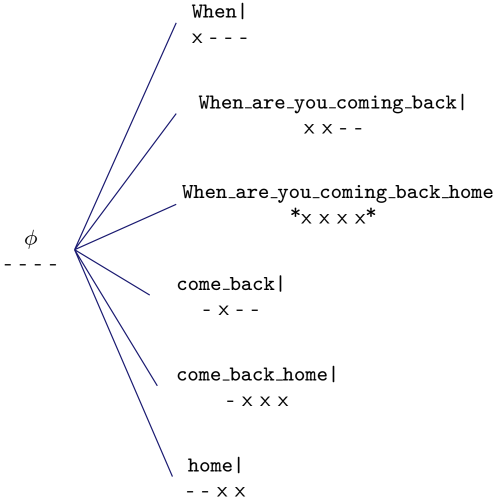
SMT, components

Decoding

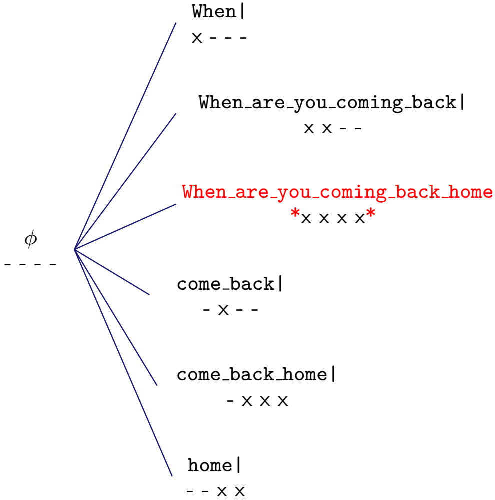
ϕ

- - - -

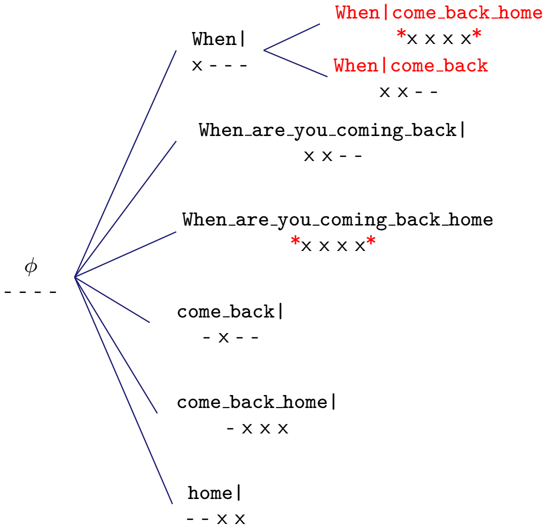
SMT, components
Decoding



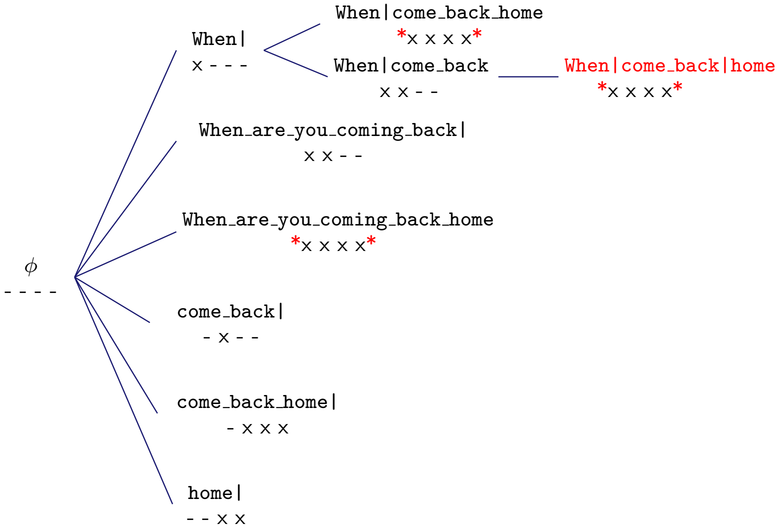
SMT, components
Decoding



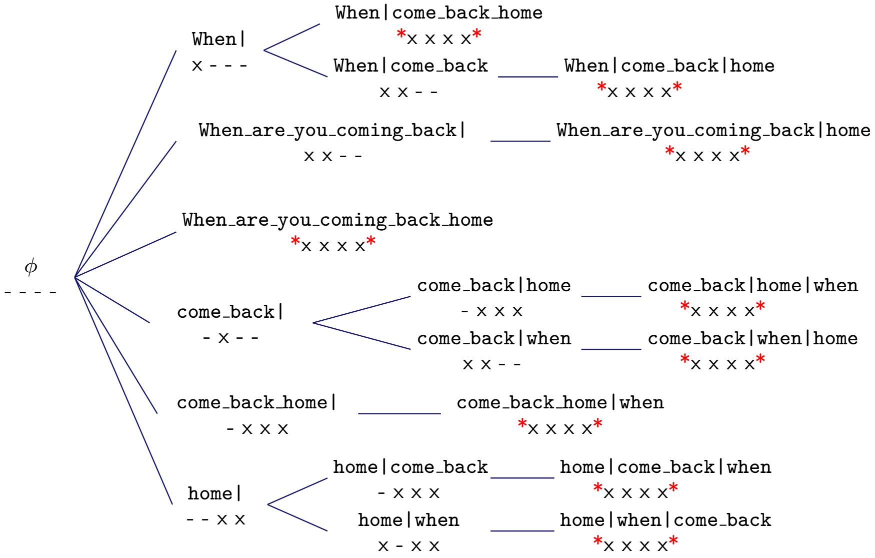
SMT, components
Decoding



SMT, components
Decoding



SMT, components
Decoding



Exhaustive search

- As a result, one should have an estimation of the cost of each hypothesis, being the lowest cost one the best translation.

But...

- The number of hypotheses is exponential with the number of source words.
  (30 words sentence $\Rightarrow$ $2^{30} = 1,073,741,824$ hypotheses!)

Solution

- Optimise the search by:
  - Hypotheses recombination
  - Beam search and pruning

Exhaustive search

- As a result, one should have an estimation of the cost of each hypothesis, being the lowest cost one the best translation.

But...

- The number of hypotheses is exponential with the number of source words.
  (30 words sentence $\Rightarrow$ $2^{30} = 1,073,741,824$ hypotheses!)

Solution

- Optimise the search by:
  - Hypotheses recombination
  - Beam search and pruning

Exhaustive search

- As a result, one should have an estimation of the cost of each hypothesis, being the lowest cost one the best translation.

But...

- The number of hypotheses is exponential with the number of source words.
  (30 words sentence $\Rightarrow$ $2^{30} = 1,073,741,824$ hypotheses!)

Solution

- Optimise the search by:
  - Hypotheses recombination
  - Beam search and pruning

Hypotheses recombination

Combine hypotheses with the same source words translated, keep that with a lower cost.

`When|come_back_home` $\Longleftrightarrow$ `When|come_back|home`

`x x x x` `x x x x`

- Risk-free operation. The lowest cost translation is still there.
- But the space of hypothesis is not reduced enough.

Hypotheses recombination

Combine hypotheses with the same source words translated, keep that with a lower cost.

```
When|come_back_home      ⟺      When|come_back|home
      x x x x                         x x x x
```

- Risk-free operation. The lowest cost translation is still there.
- But the space of hypothesis is not reduced enough.

Hypotheses recombination

Combine hypotheses with the same source words translated, keep that with a lower cost.

`When|come_back_home` ⟺ `When|come_back|home`

x x x x   x x x x

- Risk-free operation. The lowest cost translation is still there.
- But the space of hypothesis is not reduced enough.

Beam search and pruning (at last!)

Compare hypotheses with the same number of translated source words and prune out the inferior ones.

What is an inferior hypothesis?

- The quality of a hypothesis is given by the cost so far and by an estimation of the future cost.
- Future cost estimations are only approximate, so the pruning is not risk-free.

Beam search and pruning (at last!)

Strategy:

- Define a beam size (by threshold or number of hypotheses).
- Distribute the hypotheses being generated in stacks according to the number of translated source words, for instance.
- Prune out the hypotheses falling outside the beam.
- The hypotheses to be pruned are those with a higher (current + future) cost.

Decoding: keep in mind

- Standard SMT decoders translate the sentences from left to right by expanding hypotheses.
- Beam search decoding is one of the most efficient approach.
- But, the search is only approximate, so, the best translation can be lost if one restricts the search space too much.

Outline

SMT, the log-linear model
Motivation

Maximum likelihood (ML)

$$\hat{e} = \mathrm{argmax}_{\mathrm{e}} P(e|f) = \mathrm{argmax}_{\mathrm{e}}\ P(e)\, P(f|e)$$

Maximum entropy (ME)

$$\hat{e} = \mathrm{argmax}_{\mathrm{e}} P(e|f) = \mathrm{argmax}_{\mathrm{e}}\ \exp\left\{\sum \lambda_m\, h_m(f,e)\right\}$$

$$\hat{e} = \mathrm{argmax}_{\mathrm{e}} \log P(e|f) = \mathrm{argmax}_{\mathrm{e}}\ \sum \lambda_m\, h_m(f,e)$$

Log-linear model

SMT, the log-linear model

Motivation

Maximum likelihood (ML)

$$\hat{e} = \mathrm{argmax}_{\mathrm{e}} P(e|f) = \mathrm{argmax}_{\mathrm{e}} \; P(e)\, P(f|e)$$

Maximum entropy (ME)

$$\hat{e} = \mathrm{argmax}_{\mathrm{e}} P(e|f) = \mathrm{argmax}_{\mathrm{e}} \; \exp\left\{\sum \lambda_m \, h_m(f, e)\right\}$$

$$\hat{e} = \mathrm{argmax}_{\mathrm{e}} \log P(e|f) = \mathrm{argmax}_{\mathrm{e}} \; \sum \lambda_m \, h_m(f, e)$$

Log-linear model

SMT, the log-linear model

Motivation

Maximum likelihood (ML)

$$\hat{e} = \mathrm{argmax}_{\mathrm{e}} P(e|f) = \mathrm{argmax}_{\mathrm{e}}\ P(e)\, P(f|e)$$

Maximum entropy (ME)

$$\hat{e} = \mathrm{argmax}_{\mathrm{e}} P(e|f) = \mathrm{argmax}_{\mathrm{e}}\ \exp\left\{\sum \lambda_m\, h_m(f,e)\right\}$$

$$\hat{e} = \mathrm{argmax}_{\mathrm{e}} \log P(e|f) = \mathrm{argmax}_{\mathrm{e}}\ \sum \lambda_m\, h_m(f,e)$$

Log-linear model

SMT, the log-linear model

Motivation

Maximum likelihood (ML)

$$\hat{e} = \mathrm{argmax_e} P(e|f) = \mathrm{argmax_e}\; P(e)\, P(f|e)$$

Maximum entropy (ME)

$$\hat{e} = \mathrm{argmax_e} \log P(e|f) = \mathrm{argmax_e}\; \sum \lambda_m\, h_m(f,e)$$

Log-linear model with

$$h_1(f,e) = logP(e),\; h_2(f,e) = logP(f|e),\; \text{and}\; \lambda_1 = \lambda_2 = 1$$

$\Rightarrow$ Maximum likelihood model

What can be achieved with the log-linear model
(as compared to maximum likelihood model)

- Extra features h_m can be easily added...
- ... but their weight λ_m must be somehow determined.
- Different knowledge sources can be used.

Standard feature functions

Eight features are usually used: $P(e)$, $P(f|e)$, $P(e|f)$, $lex(f|e)$, $lex(e|f)$, $ph(e)$, $w(e)$ and $P_d(e, f)$.

- Language model $P(e)$
  $P(e)$: Language model probability as in ML model.
- Translation model $P(f|e)$
  $P(f|e)$: Translation model probability as in ML model.
- Translation model $P(e|f)$
  $P(e|f)$: Inverse translation model probability to be added to the generative one.

Standard feature functions

Eight features are usually used: $P(e)$, $P(f|e)$, $P(e|f)$, $lex(f|e)$, $lex(e|f)$, $ph(e)$, $w(e)$ and $P_d(e,f)$.

- Translation model $lex(f|e)$
  $lex(f|e)$: Lexical translation model probability.
- Translation model $lex(e|f)$
  $lex(e|f)$: Inverse lexical translation model probability.
- Phrase penalty $ph(e)$
  $ph(e)$: A constant cost per produced phrase.

Standard feature functions

Eight features are usually used: $P(e)$, $P(f|e)$, $P(e|f)$, $lex(f|e)$, $lex(e|f)$, $ph(e)$, $w(e)$ and $P_d(e,f)$.

- Word penalty $w(e)$
  $w(e)$: A constant cost per produced word.
- Distortion $P_d(e,f)$
  $P_d(\mathrm{ini}_{\mathrm{phrase}_i}, \mathrm{end}_{\mathrm{phrase}_{i-1}})$: Relative distortion probability distribution. A simple distortion model:
  $P_d(\mathrm{ini}_{\mathrm{phrase}_i}, \mathrm{end}_{\mathrm{phrase}_{i-1}}) = \alpha|\mathrm{ini}_{\mathrm{phrase}_i} - \mathrm{end}_{\mathrm{phrase}_{i-1}} - 1|$

SMT, components

The translation model $P(f|e)$

In practice,

```
cluster:/home/moses/model> zmore phrase-table.gz
```

```
be consistent ||| coherentes ||| 0.0384615 0.146893 0.0833333 0.0116792 2.718 ||| 1-0 ||| 26 12
be consistent ||| sean coherentes ||| 0.2 0.00022714 0.0833333 0.0916808 2.718 ||| 0-0 1-1 ||| 5 12
be consistent ||| sean consistentes ||| 0.5 0.000104834 0.0833333 0.0785835 2.718 ||| 0-0 1-1 ||| 2 12
be consistent ||| ser coherente ||| 0.5 0.0204044 0.166667 0.569957 2.718 ||| 0-0 1-1 ||| 4 12
be consistent ||| ser consecuente ||| 1 0.000340072 0.0833333 0.759942 2.718 ||| 0-0 1-1 ||| 1 12
be consistent ||| ser consistente ||| 1 0.00850183 0.5 0.633285 2.718 ||| 0-0 1-1 ||| 6 12
consistent when ||| coherente cuando se ||| 1 0.00783857 1 0.329794 2.718 ||| 0-0 1-1 1-2 ||| 1 1
consistent ||| adecuado ||| 0.00512821 0.0112994 0.00671141 0.009009 2.718 ||| 0-0 ||| 195 149
consistent ||| coherencia ||| 0.137931 0.0282486 0.0268456 0.0847458 2.718 ||| 0-0 ||| 29 149
consistent ||| constante ||| 0.0333333 0.0112994 0.0134228 0.0307692 2.718 ||| 0-0 ||| 60 149
consistent ||| constantes ||| 0.0625 0.0056497 0.00671141 0.047619 2.718 ||| 0-0 ||| 16 149
...
```

State of the art?

Software such as `Moses` makes easy the incorporation of more sophisticated reordering.

From a **distance-based** reordering
(1 feature)

to include orientation information
in a **lexicalised** reordering.
(3-6 features)

SMT, the log-linear model

Digression: lexicalised reordering or distortion

From where and how can one learn reorders?

(are, tornes, monotone)

SMT, the log-linear model

Digression: lexicalised reordering or distortion

From where and how can one learn reorders?

(coming back, tornes, swap)

SMT, the log-linear model

Digression: lexicalised reordering or distortion

From where and how can one learn reorders?

(home ?, casa ?, discontinuous)

3 new features estimated by frequency counts:
P_{monotone}, P_{swap} and $P_{\mathrm{discontinuous}}$ (6 when bidirectional).

$$P_{or.}(\mathrm{orientation}|f,e) = \frac{count(\mathrm{orientation},e,f)}{\sum_{or.} count(\mathrm{orientation},e,f)}$$

- Sparse statistics of the orientation types $\rightarrow$ smoothing.
- Several variations.

In practice,

```
cluster:/home/moses/model> zmore extract.o.gz
```

```
resumption ||| reanudacion ||| mono mono
resumption of the ||| reanudacion del ||| mono mono
resumption of the session ||| reanudacion del periodo de sesiones ||| mono mono
de la union ||| union ’ s ||| swap swap
competencia de la union ||| union ’ s competition ||| swap other
...
```

```
cluster:/home/moses/model> zmore reordering-table.wbe-msd-bidirectional-fe.gz
```

```
a resumption of the s ||| se reanudara el periodo de s ||| 0.200 0.200 0.600 0.600 0.200 0.200
resumption of the s ||| reanudacion del periodo de s ||| 0.995 0.002 0.002 0.995 0.002 0.002
the resumption of the s ||| la continuacion del periodo de s ||| 0.142 0.142 0.714 0.714 0.142 0.142
the resumption of the s ||| la reanudacion del periodo de s ||| 0.818 0.090 0.090 0.818 0.090 0.090
...
```

SMT, components

The translation model $P(f|e)$

```
cluster:/home/moses/model> wc -l *

   493,896,818 phrase-table
   493,896,818 reordering-table.wbe-msd-bidirectional-fe
```

```
cluster:/home/moses/model> ls -lkh *

-rw-r--r-- 1 emt ia 57G mar 3 14:01 phrase-table
-rw-r--r-- 1 emt ia 55G mar 3 14:08 reordering-table.wbe-msd-bidirectional-fe
```

Standard feature functions

13 features may be used:

- $P(e)$;
- $P(f|e)$, $P(e|f)$, $lex(f|e)$, $lex(e|f)$;
- $ph(e)$, $w(e)$;
- $P_{mon}(o|e,f)$, $P_{swap}(o|e,f)$, $P_{dis}(o|e,f)$,
- $P_{mon}(o|f,e)$, $P_{swap}(o|f,e)$, $P_{dis}(o|f,e)$.

Development training, weights optimisation

- Supervised training: a (small) aligned parallel corpus is used to determine the optimal weights.

$$\hat{e} = \mathrm{argmax_e} \log P(e|f) = \mathrm{argmax_e} \sum \lambda_m h_m(f, e)$$

Development training, weights optimisation

Strategies

- Generative training. Optimises ME objective function which has a unique optimum. Maximises the likelihood.
- Discriminative training only for feature weights (not models), or purely discriminative for the model as a whole. This way translation performance can be optimised.
- Minimum Error-Rate Training (MERT).

Development training, weights optimisation

Strategies

- Generative training. Optimises ME objective function which has a unique optimum. Maximises the likelihood.
- Discriminative training only for feature weights (not models), or purely discriminative for the model as a whole. This way translation performance can be optimised.
- Minimum Error-Rate Training (MERT).

Minimum Error-Rate Training

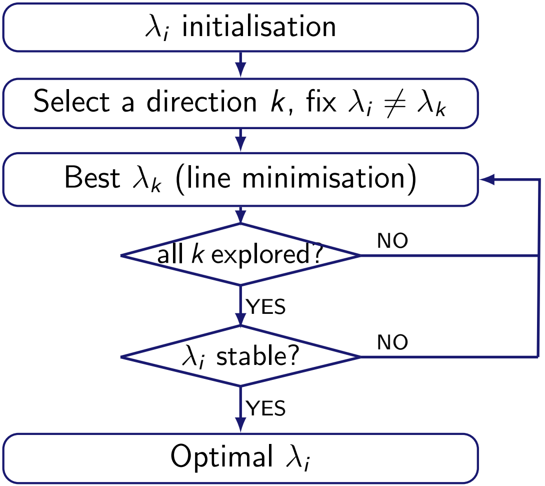
- Approach: Minimise an error function.

But... what's the error of a translation?

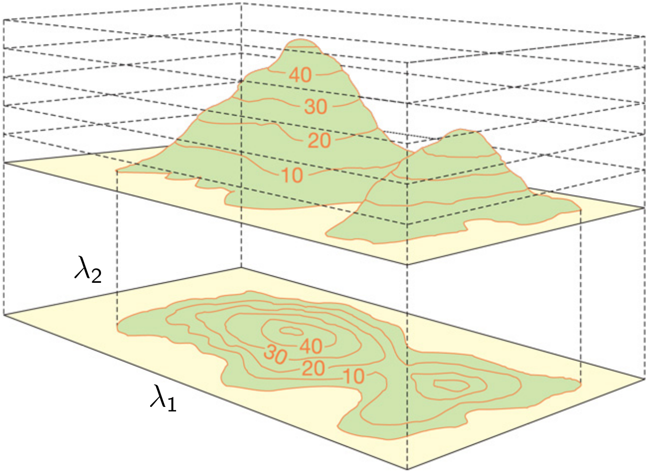
- There exist several error measures or metrics.
- Metrics not always correlate with human judgements.
- The quality of the final translation on the metric choosen for the optimisation is shown to improve.
- For the moment, let's say we use BLEU.

(More on MT Evaluation section)

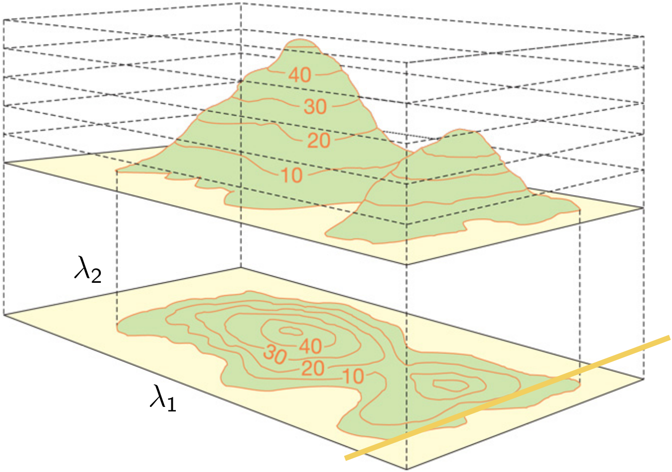
Minimum Error-Rate Training rough algorithm



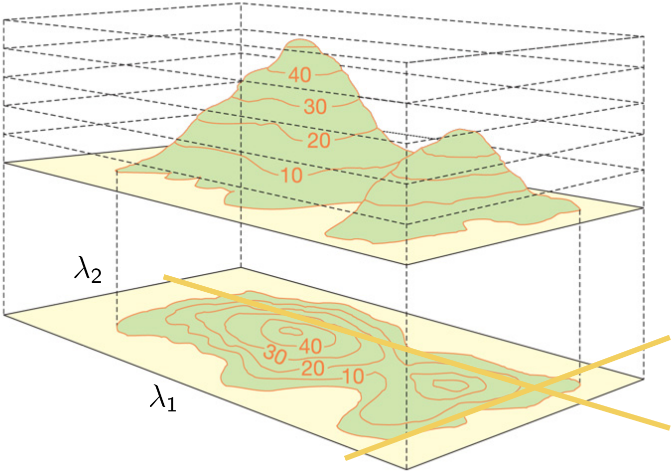
Powell's method (2D: λ_1,λ_2)



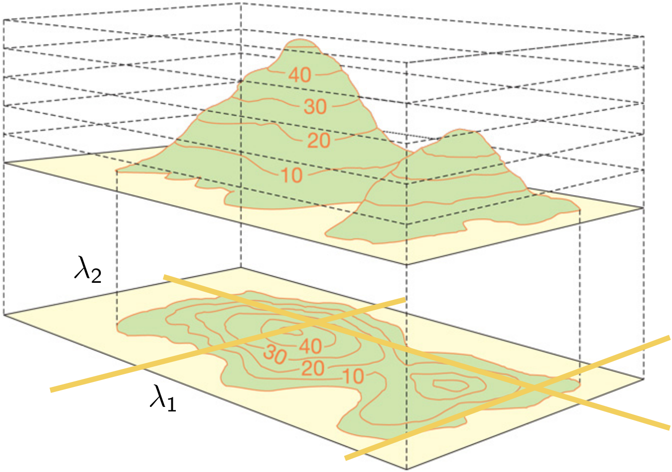
Powell's method (2D: λ_1,λ_2)



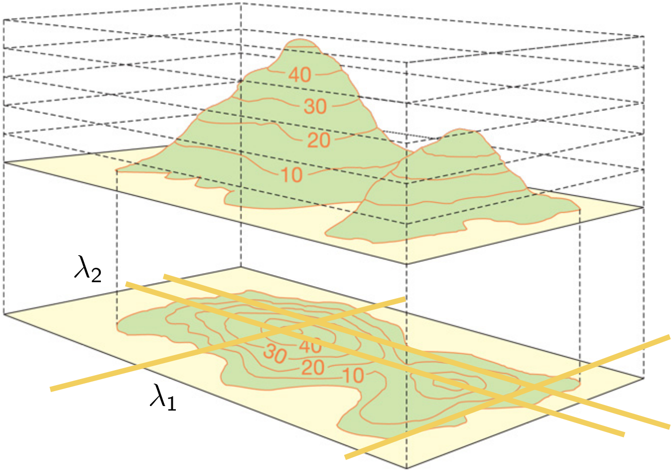
Powell's method (2D: λ_1,λ_2)



Powell's method (2D: λ_1,λ_2)



Powell's method (2D: λ_1,λ_2)



In practice,

```
# language model weights
[weight-l]
0.102111


# translation model weights
[weight-t]
0.0146796
0.0281078
0.0501881
0.087537
0.128371

# word penalty
[weight-w]
-0.142732
```

Log-linear model: keep in mind

- The log-linear model allows to include several weighted features. Standard systems use 8 (13) real features.
- The corresponding weights are optimised on a development set, a small aligned parallel corpus.
- An optimisation algorithm such as MERT is appropriate for about a dozen of features. For more features, purely discriminative learnings should be used.
- For MERT, the choice of the metric that quantifies the error in the translation is an issue.

Word alignment with...

GIZA++
`https://code.google.com/p/giza-pp`

The Berkeley Word Aligner
`https://code.google.com/p/berkeleyaligner`

Fast Align
`https://github.com/clab/fast_align`

...

Language Model with...

SRILM
http://www.speech.sri.com/projects/srilm

IRSTLM
http://sourceforge.net/projects/irstlm

RandLM
http://sourceforge.net/projects/randlm

KenLM
http://kheafield.com/code/kenlm

...

Try parameter optimisation with...

MERT
Minimum error rate training, Och (2003)

PRO
Pairwise ranked optimization, Hopkins and May (2011)

MIRA
Margin Infused Relaxed Algorithm, Hasler et al. (2011)

...

Decoding with...

Moses
`http://www.statmt.org/moses`

Phrasal
`http://nlp.stanford.edu/software/phrasal`

...

Docent
`https://github.com/chardmeier/docent`

Build your own SMT system

1. Language model with `SRILM`.
   http://www-speech.sri.com/projects/srilm/download.html
2. Word alignments with `GIZA++`.
   http://code.google.com/p/giza-pp/downloads/list
3. And everything else with the `Moses` package.
   https://github.com/moses-smt/mosesdecoder

1. Download and prepare your data

1. Parallel corpora and some tools can be downloaded for instance from the WMT 2013 web page: http://www.statmt.org/wmt13/translation-task.html

How to construct a baseline system is also explained there: http://www.statmt.org/wmt10/baseline.html

We continue with the Europarl corpus Spanish-to-English.

1. Download and prepare your data (cont'd)

2. Tokenise the corpus with WMT10 scripts.
(training corpus and development set for MERT)

```
wmt10scripts/tokenizer.perl -l es < eurov4.es-en.NOTOK.es >
eurov4.es-en.TOK.es
wmt10scripts/tokenizer.perl -l en < eurov4.es-en.NOTOK.en >
eurov4.es-en.TOK.en

wmt10scripts/tokenizer.perl -l es < eurov4.es-en.NOTOK.dev.es >
eurov4.es-en.TOK.dev.es
wmt10scripts/tokenizer.perl -l en < eurov4.es-en.NOTOK.dev.en >
eurov4.es-en.TOK.dev.en
```

1. **Download and prepare your data** (cont'd)

3. Filter out long sentences with `Moses` scripts.
   (Important for GIZA++)

```
bin/moses-scripts/training/clean-corpus-n.perl eurov4.es-en.TOK es
en eurov4.es-en.TOK.clean 1 100
```

4. Lowercase training and development with WMT10 scripts.
   (Optional but recommended)

```
wmt10scripts/lowercase.perl < eurov4.es-en.TOK.clean.es >
eurov4.es-en.es
wmt10scripts/lowercase.perl < eurov4.es-en.TOK.clean.en >
eurov4.es-en.en
```

2. Build the language model

1. Run `SRILM` on the English part of the parallel corpus or on a monolingual larger one.
(tokenise and lowercase in case it is not)

```
ngram-count -order 5 -interpolate -kndiscount -text
eurov4.es-en.en -lm eurov4.en.lm
```

3. Train the translation model

1. Use the Moses script `train-model.perl`
   This script performs the whole training:

```
train-model.perl -help

Train Phrase Model
Steps:  (--first-step to --last-step)
(1) prepare corpus
(2) run GIZA
(3) align words
(4) learn lexical translation
(5) extract phrases
(6) score phrases
(7) learn reordering model
(8) learn generation model
(9) create decoder config file
```

3. Train the translation model (cont'd)

1. So, it takes a few arguments (and a few time!):

   `moses-scripts/training/train-model.perl` **-scripts-root-dir** `bin/moses-scripts/` **-root-dir** `working-dir` **-corpus** `eurov4.es-en` **-f** `es` **-e** `en` **-alignment** `grow-diag-final-and` **-reordering** `msd-bidirectional-fe` **-lm** `0:5:eurov4.en.lm:0`

   It generates a configuration file `moses.ini` needed to run the decoder where all the necessary files are specified.

4. Tuning of parameters with MERT

1. Run the Moses script `mert-moses.pl`
   (Another slow step!)

```
moses-scripts/training/mert-moses.pl eurov4.es-en.dev.es
eurov4.es-en.dev.en mosesdecoder/bin/moses ./model/moses.ini
--working-dir ./tuning --rootdir bin/moses-scripts/
```

2. Insert weights into configuration file with WMT10 script:

```
wmt10scripts/reuse-weights.perl ./tuning/moses.ini <

./model/moses.ini > moses.weight-reused.ini
```

5. Run `Moses` decoder on a test set

1. Tokenise and lowecase the test set as before.
2. Filter the model with `Moses` script.
   (mandatory for large translation tables)

   ```
   moses-scripts/training/filter-model-given-input.pl ./filteredmodel
   moses.weight-reused.ini testset.es
   ```

3. Run the decoder:

   ```
   mosesdecoder/bin/moses -f ./filteredmodel/moses.ini < testset.es >
   testset.translated.en
   ```