Introduction to Statistical Machine Translation

Cristina España-Bonet
DFKI GmbH

Summer Semester 2023
25th April & 2nd, 8th May 2023

Overview

1. Introduction
2. Basics
3. Components
4. The log-linear model
5. Beyond standard SMT

Part I: SMT background

Overview

Part II: SMT Lab

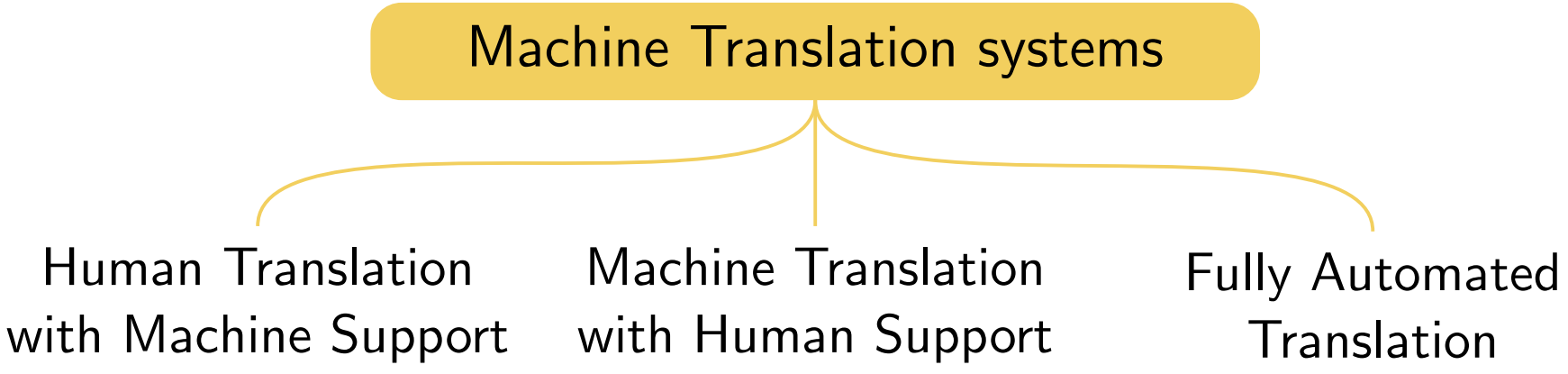
Part I

SMT background

Outline

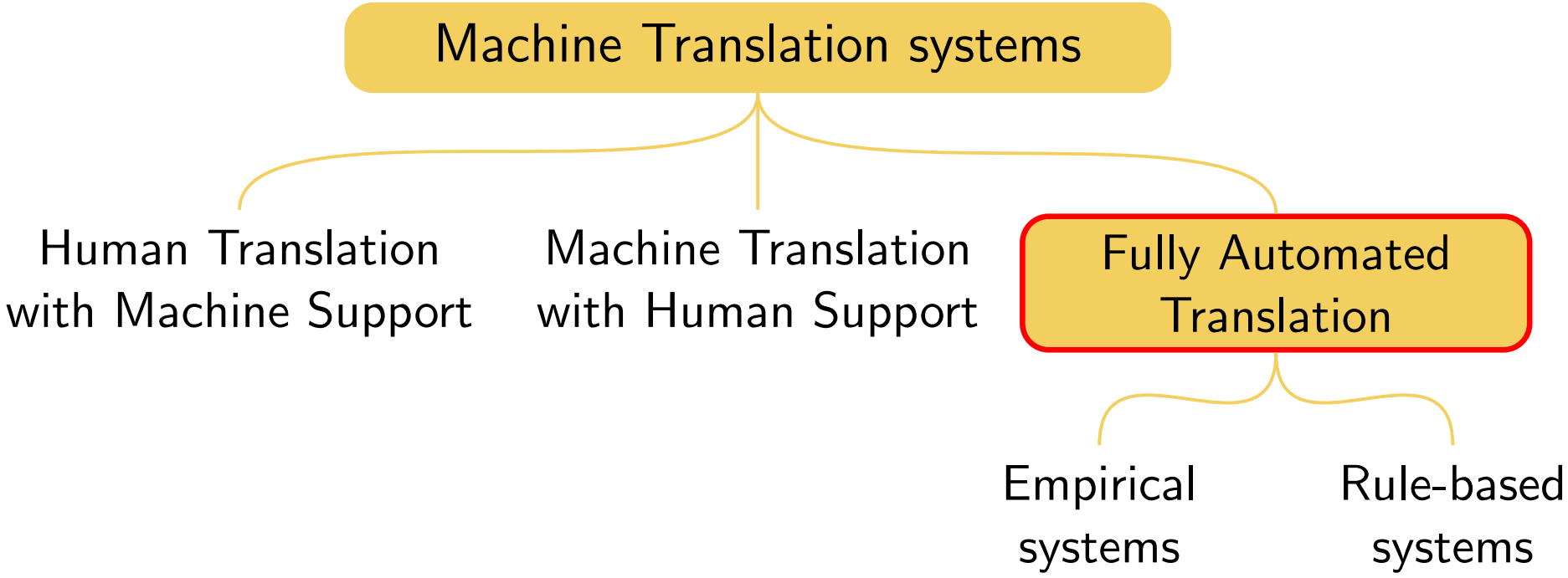
Introduction

Machine Translation Taxonomy



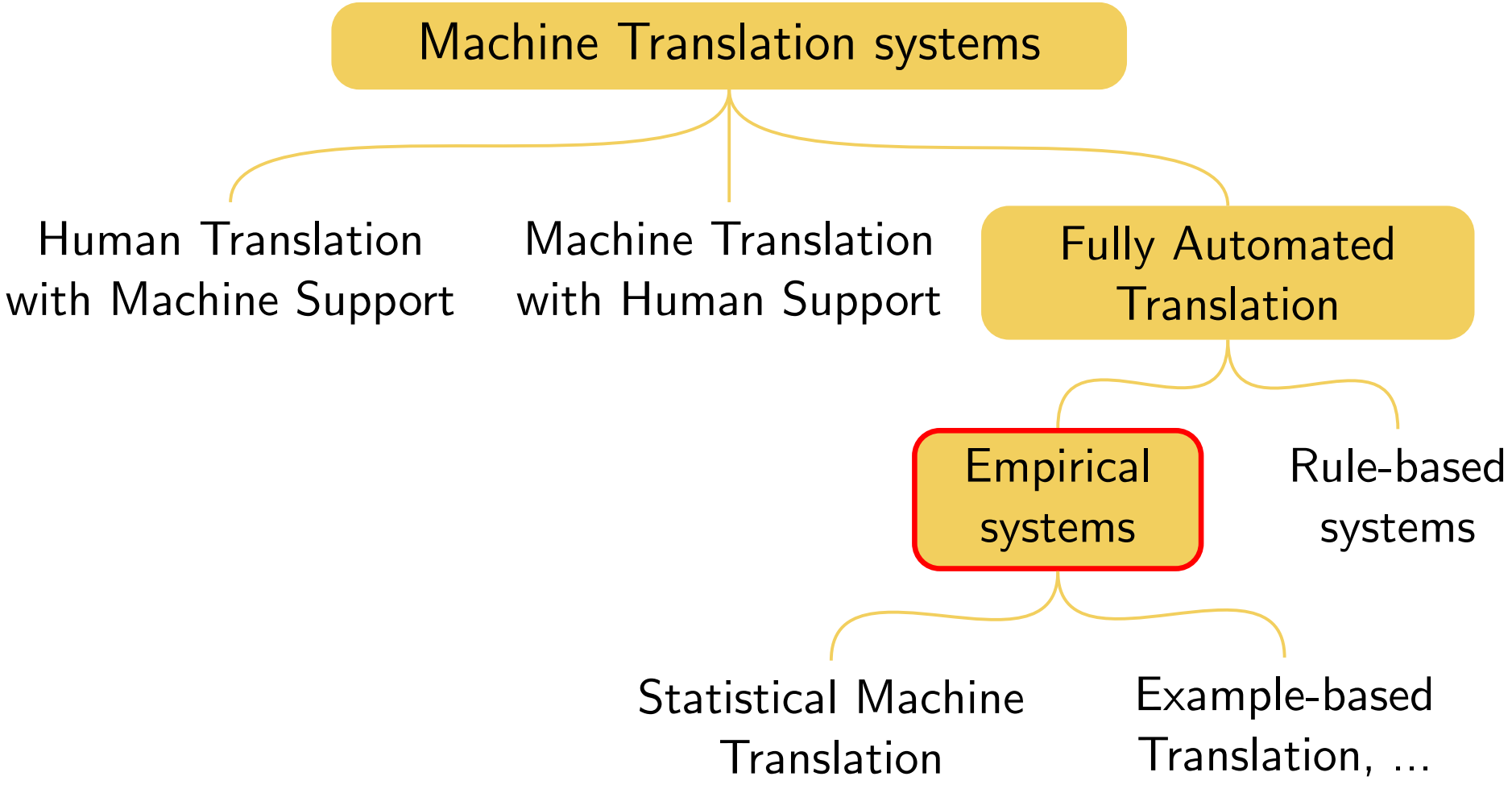
Introduction

Machine Translation Taxonomy



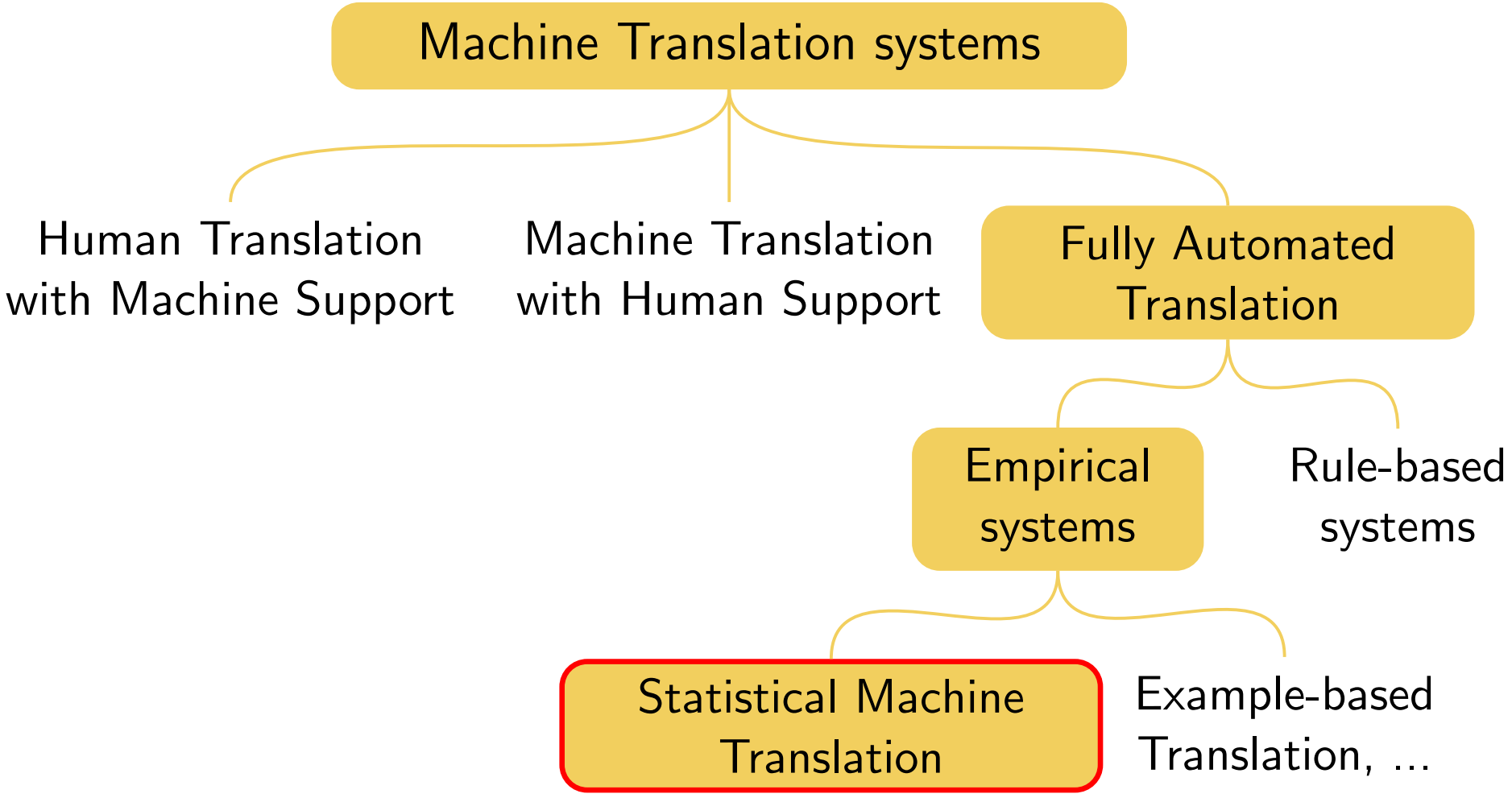
Introduction

Machine Translation Taxonomy



Introduction

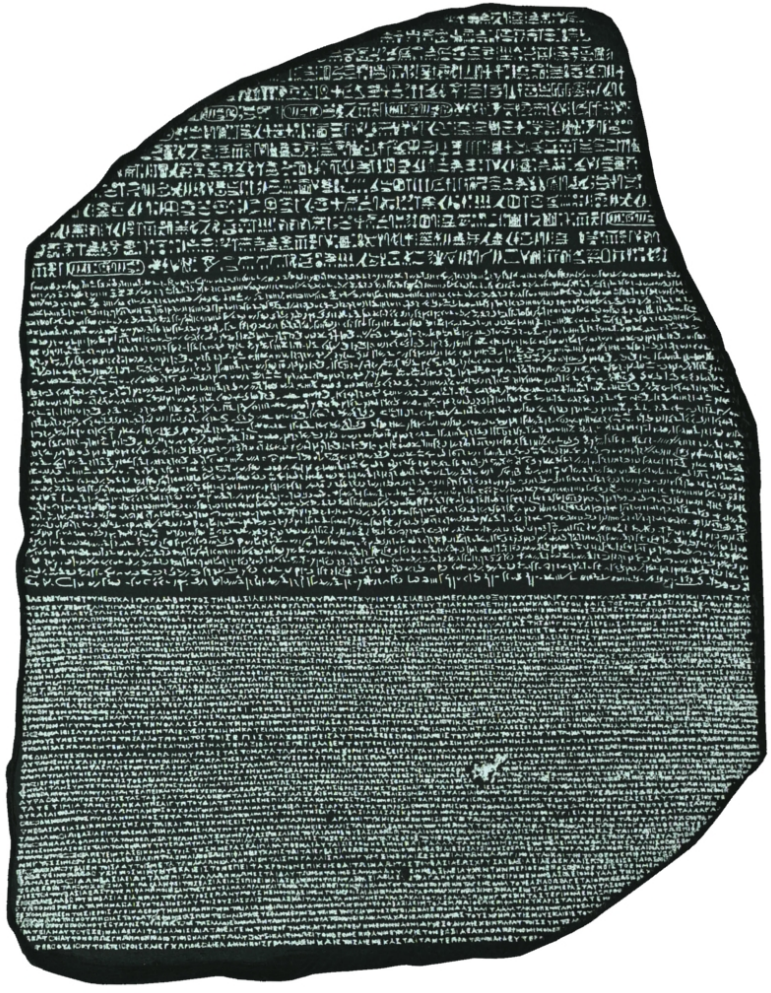
Machine Translation Taxonomy



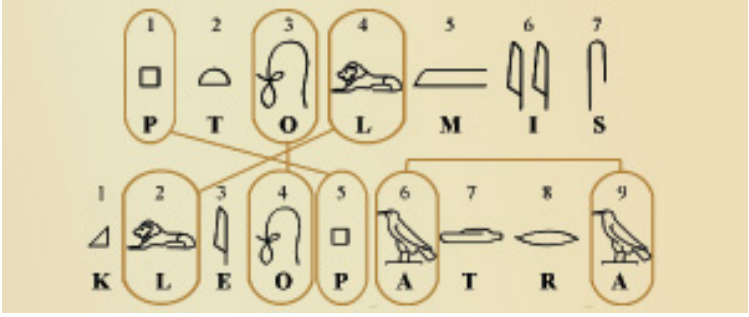
Introduction

Empirical Machine Translation

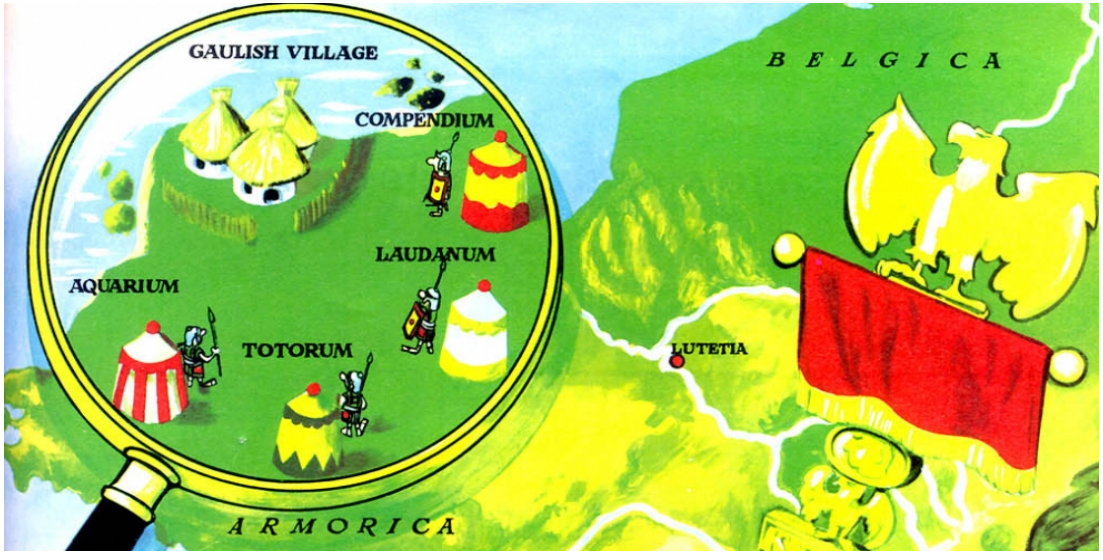
Empirical MT relies on aligned corpora



Empirical MT relies on aligned corpora



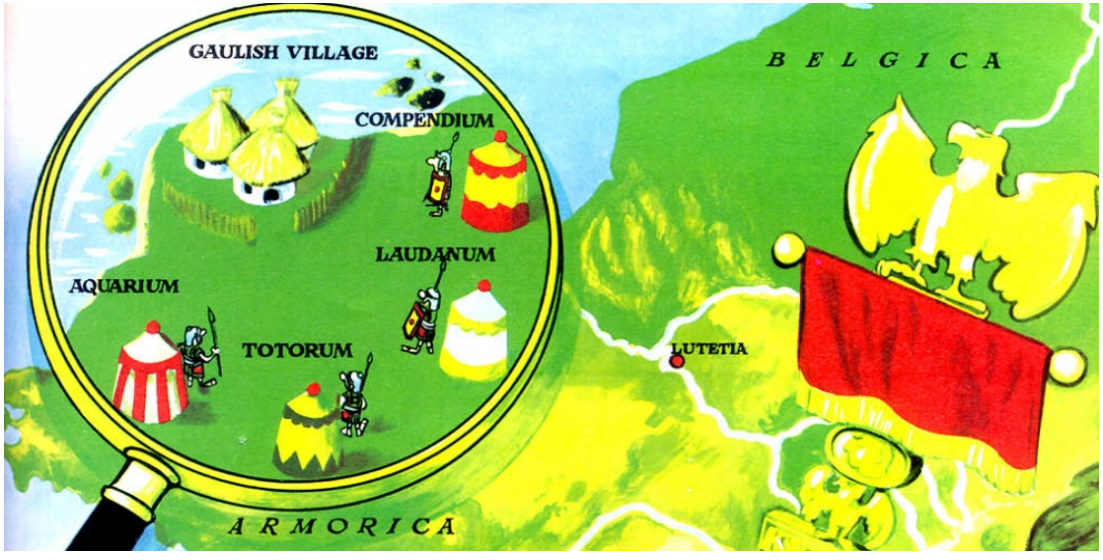
Empirical MT relies on large parallel aligned corpora



Som a l'any 50 abans de Crist. Tota la Gàl·lia és ocupada pels romans... Tota? No! Un llogaret del Nord habitat per gals indomables rebutja una i altra vegada ferotgement l'invassor. La vida doncs no és gens planera per als legionaris romans dels petits campaments de Babaòrum, Aquàrium, Laundànum i Petibònum...

The year is 50 B.C. Gaul is entirely occupied by the Romans. Well, not entirely... One small village of indomitable Gauls still holds out against the invaders. And life is not easy for the Roman legionaries who garrison the fortified camps of Totorum, Aquarium, Laudanum and Compendium...

Empirical MT relies on large parallel aligned corpora



Som a l'any 50 abans de Crist. Tota la Gàl·lia és ocupada pels romans... Tota? No! Un llogaret del Nord habitat per gals indomables rebutja una i altra vegada ferotgement l'invassor. La vida doncs no és gens planera per als legionaris romans dels petits campaments de Babaòrum, Aquàrium, Laundànum i Petibònum...

The year is 50 B.C. Gaul is entirely occupied by the Romans. Well, not entirely... One small village of indomitable Gauls still holds out against the invaders. And life is not easy for the Roman legionaries who garrison the fortified camps of Totorum, Aquarium, Laudanum and Compendium...

Empirical MT relies on large parallel aligned corpora

Som a l'any 50 abans de Crist. Tota la Gàl·lia és ocupada pels romans... Tota? No! Un llogaret del Nord habitat per gals indomables rebutja una i altra vegada ferotgement l'invassor. La vida doncs no és gens planera per als legionaris romans dels petits campaments de Babaòrum, Aquàrium, Laundànum i Petibònum...

Astèrix. És l'heroic petit guerrer d'aquestes aventures, viu com una centella i enginyosament astut. Per això sempre li són encomanades les missions més perilloses. Extrau la seva terrorífica força de la beguda màgica inventada pel druida Panoràmix.

Obèlix. És l'antic inseparable d'Astèrix. Fa de repartidor de menhirs i li agrada d'allò més la carn de porc senglar. És capaç d'abandonar-ho tot per tal de seguir Astèrix en una nova aventura. Sobretot si no hi manquen els senglars i fortes batusses.

Copdegarròtix. És el cap de la tribu. Majestuós, valent i desconfiat alhora, el vell guerrer ès respectat pels seus homes i temut pels seus enemics. Tan sols una cosa li fa por: que el cel li pugui caure damunt del cap! Però, tal com ell mateix acostuma a dir, "Qui dia passa, any empeny!".

The year is 50 B.C. Gaul is entirely occupied by the Romans. Well, not entirely... One small village of indomitable Gauls still holds out against the invaders. And life is not easy for the Roman legionaries who garrison the fortified camps of Totorum, Aquarium, Laudanum and Compendium...

Asterix, the hero of these adventures. A shrewd, cunning little warrior; all perilous missions are immediately entrusted to him. Asterix gets his superhuman strength from the magic potion brewed by the druid Getafix...

Obelix, Asterix's inseparable friend. A menhir delivery-man by trade; addicted to wild boar. Obelix is always ready to drop everything and go off on a new adventure with Asterix - so long as there's wild boar to eat, and plenty of fighting.

Finally, Vitalstitistix, the chief of the tribe. Majestic, brave and hot-tempered, the old warrior is respected by his men and feared by his enemies. Vitalstitistix himself has only one fear; he is afraid the sky may fall on his head tomorrow. But as he always says, "Tomorrow never comes".

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	$10.7 \cdot 10^6$	$300 \cdot 10^6$

Books

Title	# words (approx.)
The Bible	$0.8 \cdot 10^6$
The Dark Tower series	$1.2 \cdot 10^6$
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	$10.7 \cdot 10^6$	$300 \cdot 10^6$

Books

Title	# words (approx.)
The Bible	$0.8 \cdot 10^6$
The Dark Tower series	$1.2 \cdot 10^6$
Encyclopaedia Britannica	$44 \cdot 10^6$

In practice

WMT14 parallel data

Corpus	# segments	# tokens
Europarl ENG	1,928,274	52,048,855
Europarl SPA	1,928,274	53,996,661
News Commentary ENG	155,615	3,901,839
News Commentary SPA	155,615	4,364,802
United Nations ENG	10,749,388	283,672,192
United Nations SPA	10,749,388	318,045,340
Total (ENG+SPA)	25,666,554	716,029,689

http://www.statmt.org/wmt14/translation-task.html

In practice

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

- Data: `www.statmt.org/wmt14/`
- Software: `SRILM`, `GIZA++` & `Moses`

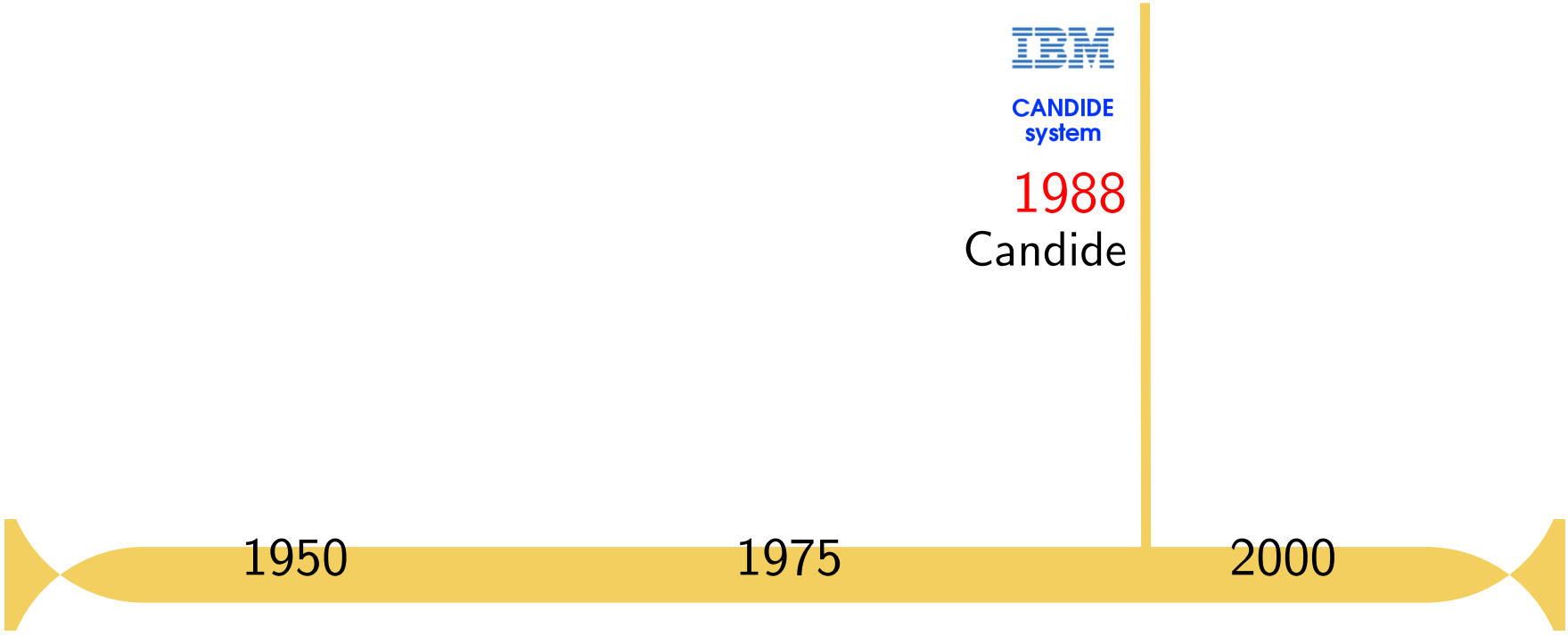
Standard tools, but not exclusive

Use it for the lab!

Outline

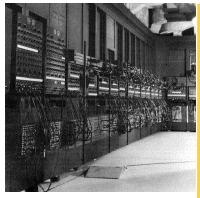
SMT, basics

The beginnings, summarised timeline

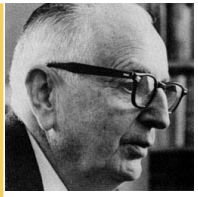


SMT, basics

The beginnings, summarised timeline



1946
ENIAC



1949
Weaver
memo

1966
Alpac report

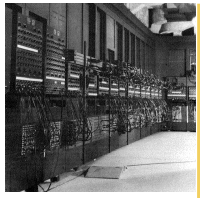
IBM
CANDIDE
system

1988
Candide

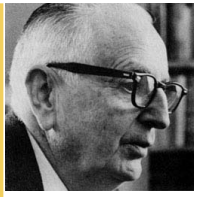
1950 1975 2000

SMT, basics

The beginnings, summarised timeline



1946
ENIAC



1949
Weaver
memo

1966
Alpac report

IBM
CANDIDE
system

1988
Candide

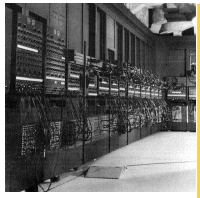
1950 1975 2000

Dictionary
MT systems

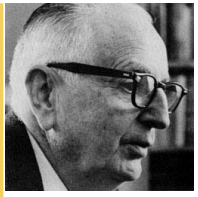
Rule-based
MT systems

SMT, basics

The beginnings, summarised timeline



1946
ENIAC



1949
Weaver
memo

1966
Alpac report

IBM
CANDIDE
system

1988
Candidate

2001
BLEU

1950 1975 2000

Dictionary
MT systems

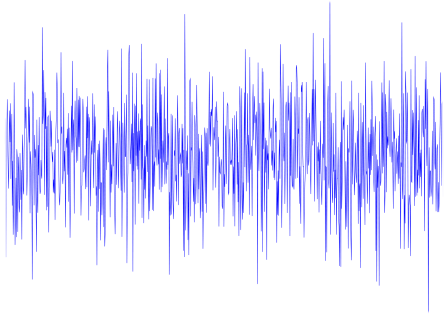
Rule-based
MT systems

Empirical
MT systems

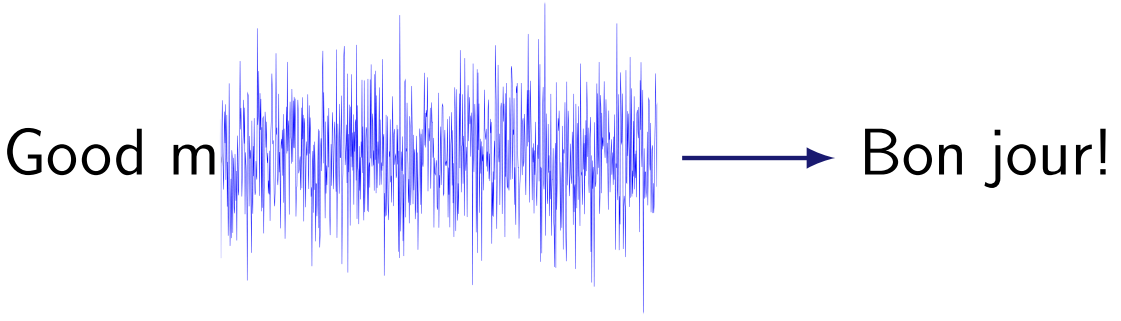
SMT, basics

The Noisy Channel approach

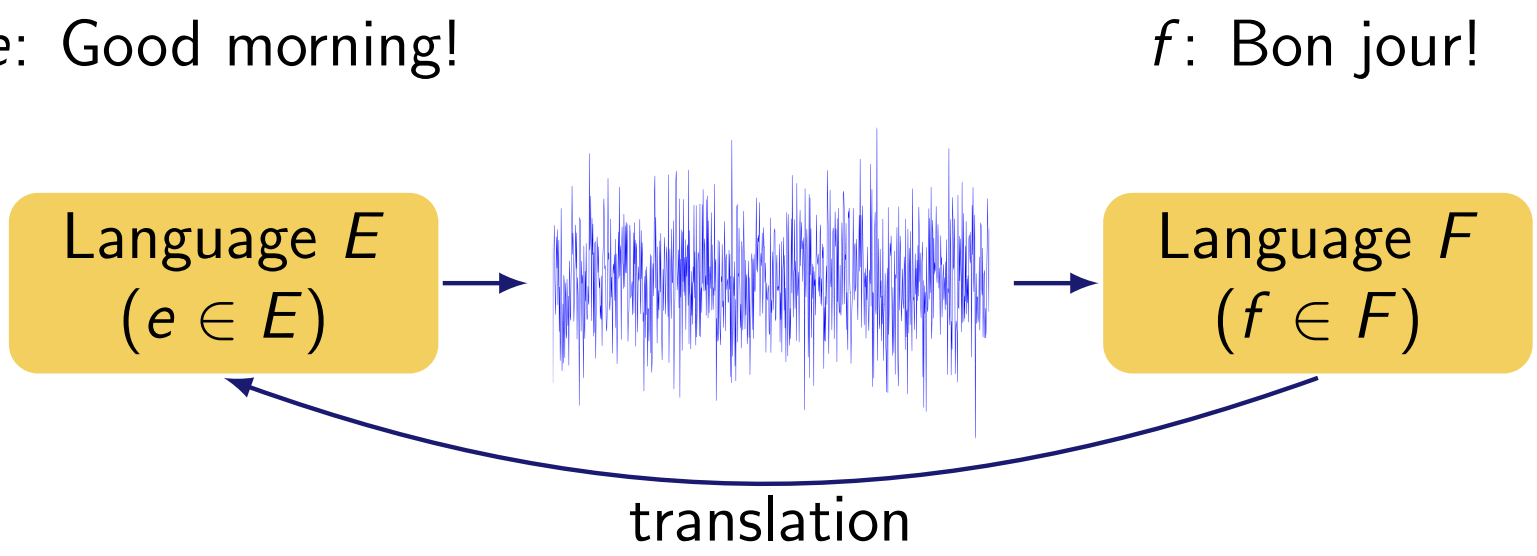
The Noisy Channel as a statistical approach to translation:

Good morning! → 

The Noisy Channel as a statistical approach to translation:

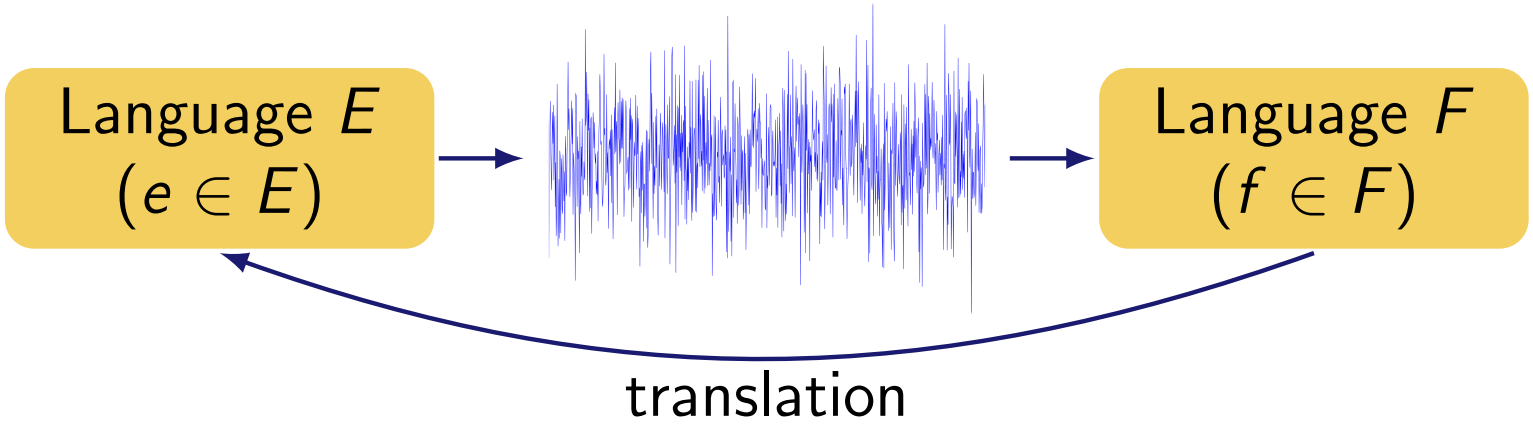


The Noisy Channel as a statistical approach to translation:



SMT, basics

The Noisy Channel approach

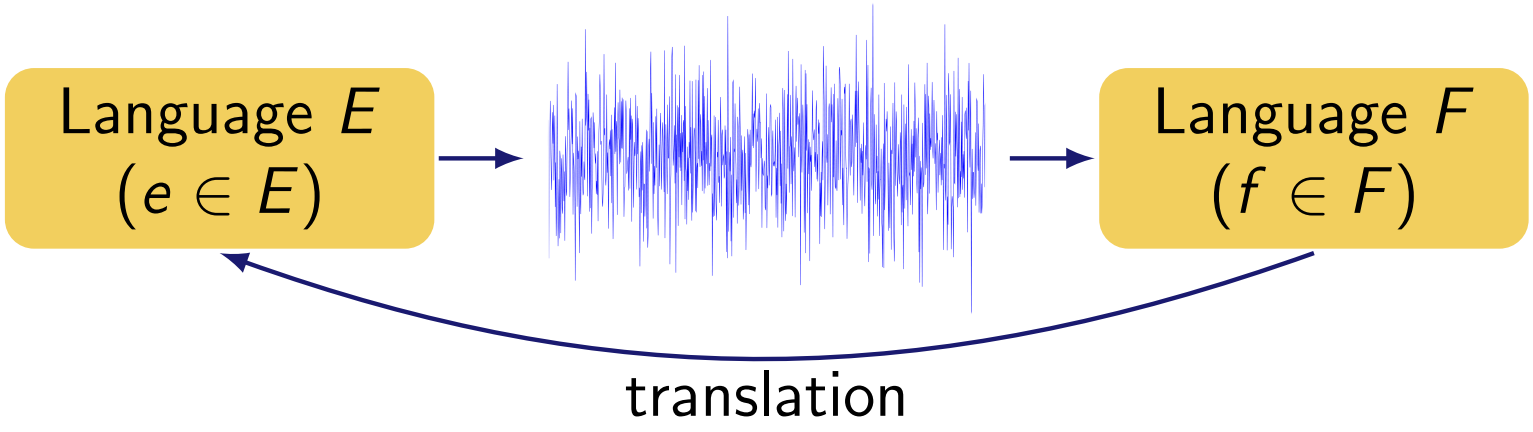


Mathematically:

$$P(e|f)$$

SMT, basics

The Noisy Channel approach



Mathematically:

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

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

SMT, basics

Components

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

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

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

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

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

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*

Outline

SMT, components

The language model $P(e)$

- We are talking about **statistical** language modelling
  - LM in general deserves a full course
  - GPT and BERT (and relatives) are **neural** language models
- SMT can include a neural LM as well
- Many more examples (especially on discounting) on Philipp Koehn's slides
  - `http://www2.statmt.org/book/`

SMT, components

The language model $P(e)$

Language model

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

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!

SMT, components

The language model $P(e)$

Language model

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

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!

SMT, components

The language model $P(e)$

Language model

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

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`

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

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`

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

where, for each factor,

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

SMT, components

The language model $P(e)$

Example, a 4-gram model

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

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

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,

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

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,

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

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,

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

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,

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

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,

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

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,

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

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,

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

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,

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

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,

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

SMT, components

The language model $P(e)$

4-gram conditions on a lot of context, so given sufficient training data these counts will be high and it will converge to the *true value*

4-gram many counts will be equal to zero, so we need many samples to get a good estimate

1-gram will converge relatively quickly to their expected value, and so don't need many samples

1-gram completely ignores context, and so it will converge to a less-good estimator as the number of training samples increase

SMT, components

The language model $P(e)$

4-gram conditions on a lot of context, so given sufficient training data these counts will be high and it will converge to the *true value*

4-gram many counts will be equal to zero, so we need many samples to get a good estimate

1-gram will converge relatively quickly to their expected value, and so don't need many samples

1-gram completely ignores context, and so it will converge to a less-good estimator as the number of training samples increase

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
- Discounting

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
- Discounting

Smoothing techniques

- Discounting
  - Keep part of the probabilty mass for unseen words
- Back-off models
  - If n-gram is not present, go to $(n-1)$-gram
- Linear interpolation

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

Smoothing techniques

- Discounting
  - Keep part of the probabilty mass for unseen words
- Back-off models
  - If n-gram is not present, go to $(n-1)$-gram
- Linear interpolation

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

Smoothing techniques

- Discounting
  - Keep part of the probabilty mass for unseen words
- Back-off models
  - If n-gram is not present, go to $(n-1)$-gram
- Linear interpolation

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

Smoothing techniques

- Discounting
  - Keep part of the probabilty mass for unseen words
- Back-off models
  - If n-gram is not present, go to $(n-1)$-gram
- Linear interpolation

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

SMT, components

The language model $P(e)$

Example 1: 3-gram language model with linear interpolation

- Consider the weights

$$\lambda_1 = \lambda_2 = \lambda_3 = 1/3;\ \lambda_0 = 0$$

- Consider the corpus
  - the green book STOP
  - my blue book STOP
  - his green house STOP
  - book STOP

SMT, components

The language model $P(e)$

Example 1: 3-gram language model with linear interpolation

$P(\text{book}|\text{the, green})$?

$P(\text{book}|\text{the, green}) = 0.571$

$$\frac{1}{3}P(\text{book}|\text{the, green}) + \frac{1}{3}P(\text{book}|\text{green}) + \frac{1}{3}P(\text{book}) =$$

$$\frac{1}{3}\frac{\text{Count(the,green,book)}}{\text{Count(the,green)}} + \frac{1}{3}\frac{\text{Count(green,book)}}{\text{Count(green)}} + \frac{1}{3}\frac{\text{Count(book)}}{\text{Count()}} =$$

$$\frac{1}{3} \times \frac{1}{1} + \frac{1}{3} \times \frac{1}{2} + \frac{1}{3} \times \frac{3}{14}$$

SMT, components

The language model $P(e)$

Example 1: 3-gram language model with linear interpolation

$P(\text{book}|\text{the, green})$?

$P(\text{book}|\text{the, green}) = 0.571$

$$\frac{1}{3}P(\text{book}|\text{the, green}) + \frac{1}{3}P(\text{book}|\text{green}) + \frac{1}{3}P(\text{book}) =$$

$$\frac{1}{3}\frac{\text{Count(the,green,book)}}{\text{Count(the,green)}} + \frac{1}{3}\frac{\text{Count(green,book)}}{\text{Count(green)}} + \frac{1}{3}\frac{\text{Count(book)}}{\text{Count()}} =$$

$$\frac{1}{3} \times \frac{1}{1} + \frac{1}{3} \times \frac{1}{2} + \frac{1}{3} \times \frac{3}{14}$$

Example 2: Discounting
(Back-Off with Good-Turing Smoothing)

x	Count(x)	P(wi|wi-1)	Count*(x)	P*(wi|wi-1)
the	48			
the, dog	15	15/48		
the, woman	11	11/48		
the, man	10	10/48		
the, park	5	5/48		
the, job	2	2/48		
the, telescope	1	1/48		
the, manual	1	1/48		
the, afternoon	1	1/48		
the, country	1	1/48		
the, street	1	1/48		

SMT, components

The language model $P(e)$

Example 2: Discounting

Discounted counts:

$$\text{Count}^*(x) = \text{Count}(x) - \text{constant}$$

Example 2: Discounting
(Back-Off with Good-Turing Smoothing)

x	Count(x)	P(wi|wi-1)	Count*(x)	P*(wi|wi-1)
the	48			
the, dog	15	15/48	14.5	14.5/48
the, woman	11	11/48	10.5	10.5/48
the, man	10	10/48	9.5	9.5/48
the, park	5	5/48	4.5	4.5/48
the, job	2	2/48	1.5	1.5/48
the, telescope	1	1/48	0.5	0.5/48
the, manual	1	1/48	0.5	0.5/48
the, afternoon	1	1/48	0.5	0.5/48
the, country	1	1/48	0.5	0.5/48
the, street	1	1/48	0.5	0.5/48

Example 2: Discounting

Discounted counts:

$$\text{Count*(x)=Count(x)-constant}$$

Left-over probability mass:

$$d(w_{i-1}) = 1 - \sum_w \frac{\text{Count}^*(\text{w}_{i-1},\text{w})}{\text{Count}(\text{w}_{i-1})}$$

In the example: $d(\text{the}) = 1 - \frac{43}{48} \sim 1 - 0.90 \sim 0.1$

SMT, components

The language model $P(e)$

Smoothing very important for good performance.
See a good summary/overview:

- Philipp Koehn's slides(/book)
  - `http://www2.statmt.org/book/`
- Daniel Jurafsky's book
  - `https://web.stanford.edu/~jurafsky/slp3/3.pdf`
- In the lab we will use a modified Kneser-Ney smoothing, have a look!

SMT, components

The language model $P(e)$

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
```

SMT, components

The language model $P(e)$

The ARPA format for LMs:

- log probabilities in base 10
- 1-grams are unconditional probabilities
  (with back-off weights)
- n-grams with $n > 1$ are conditional probabilities
  (back-off weights only sometimes)

SMT, components

The language model $P(e)$

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.

Wait!

SMT, components

The translation model $P(f|e)$

Translation model

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

Estimation of the lexical correspondence between languages.

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

SMT, components

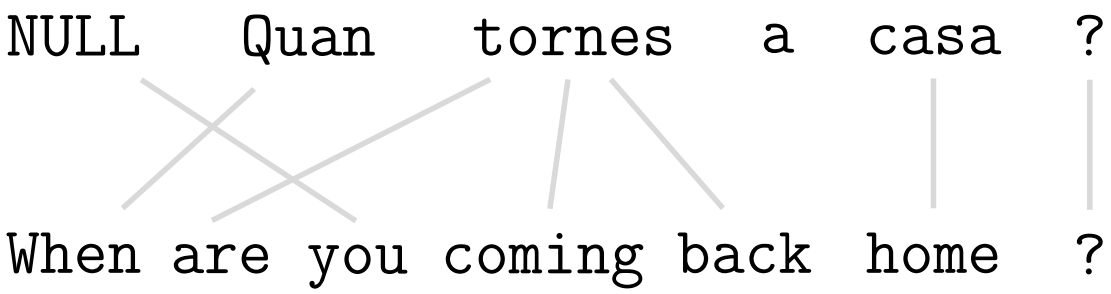
The translation model $P(f|e)$

Translation model

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

Estimation of the lexical correspondence between languages.

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



SMT, components

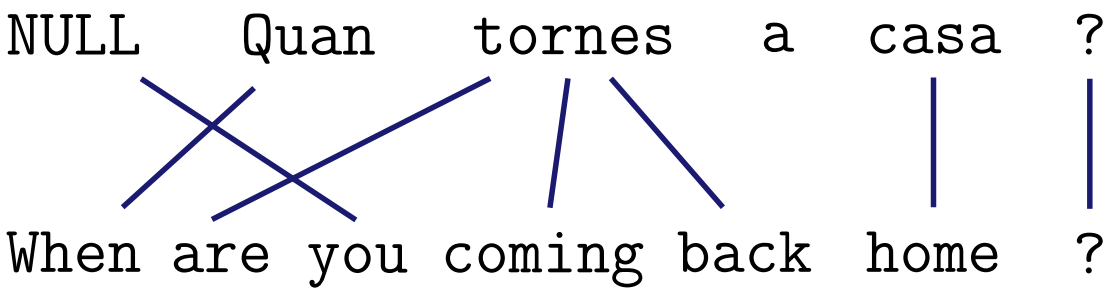
The translation model $P(f|e)$

Translation model

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

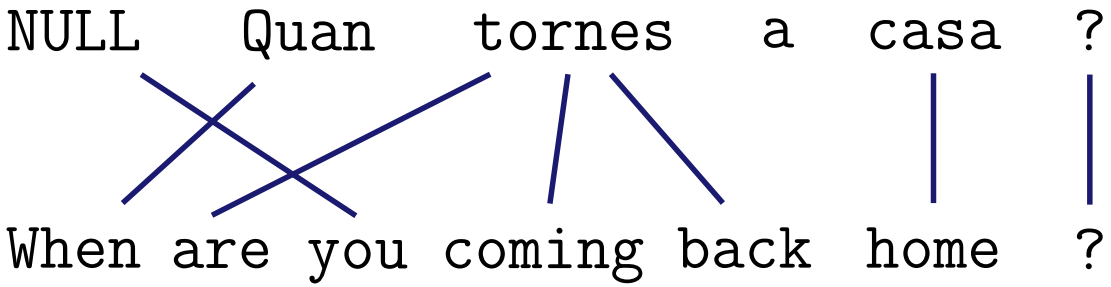
Estimation of the lexical correspondence between languages.

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.

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.

- Lexical probability t
  $t(\texttt{Quan}|\texttt{When})$: the prob. that `Quan` translates into `When`.
- Fertility n
  $n(3|\texttt{tornes})$: the prob. that `tornes` generates 3 words.

SMT, components

The translation model $P(f|e)$

Word-based models: the IBM models

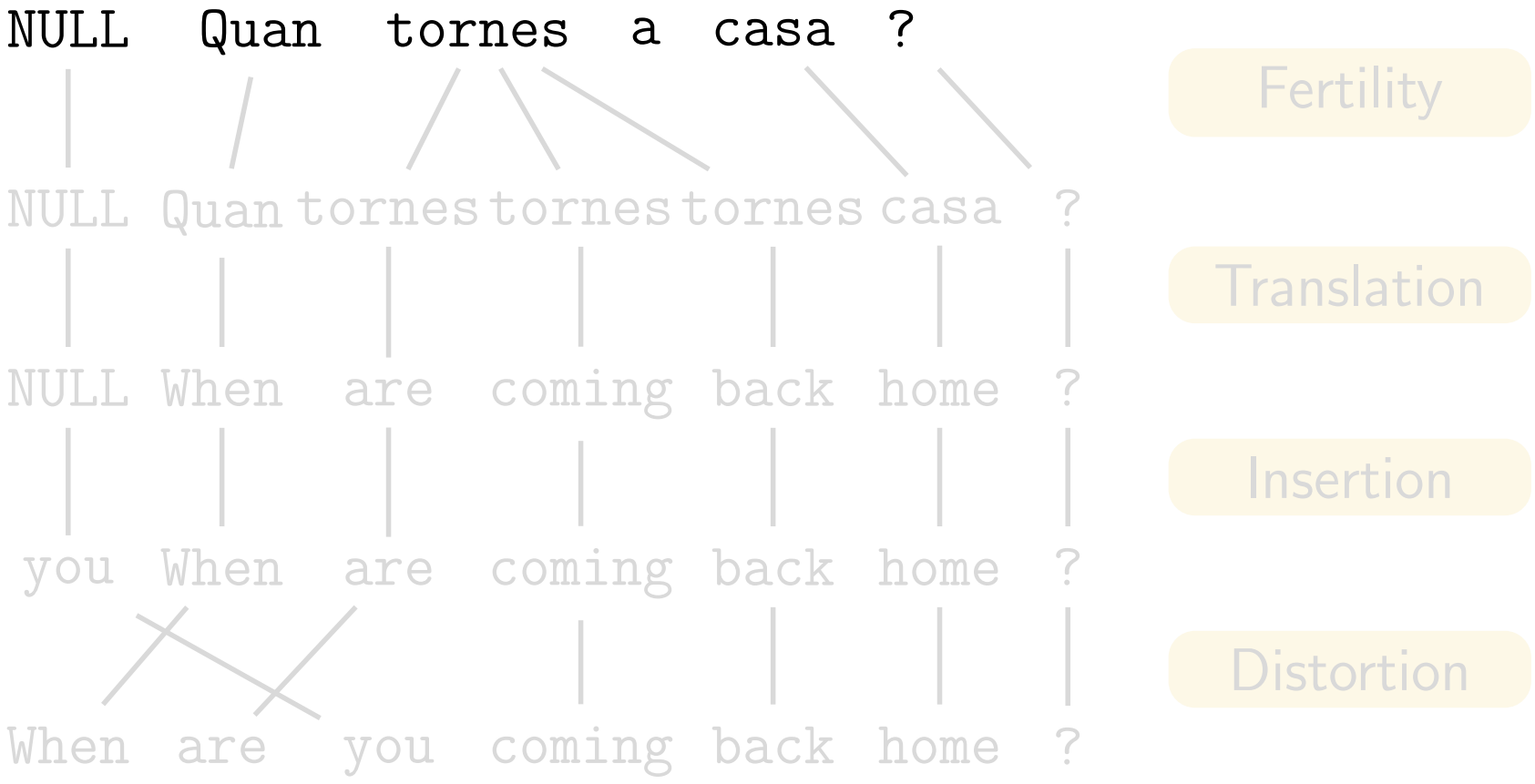
They characterise $P(f|e)$ with 4 parameters: t, n, d and p_1.

- 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`).

SMT, components

The translation model $P(f|e)$

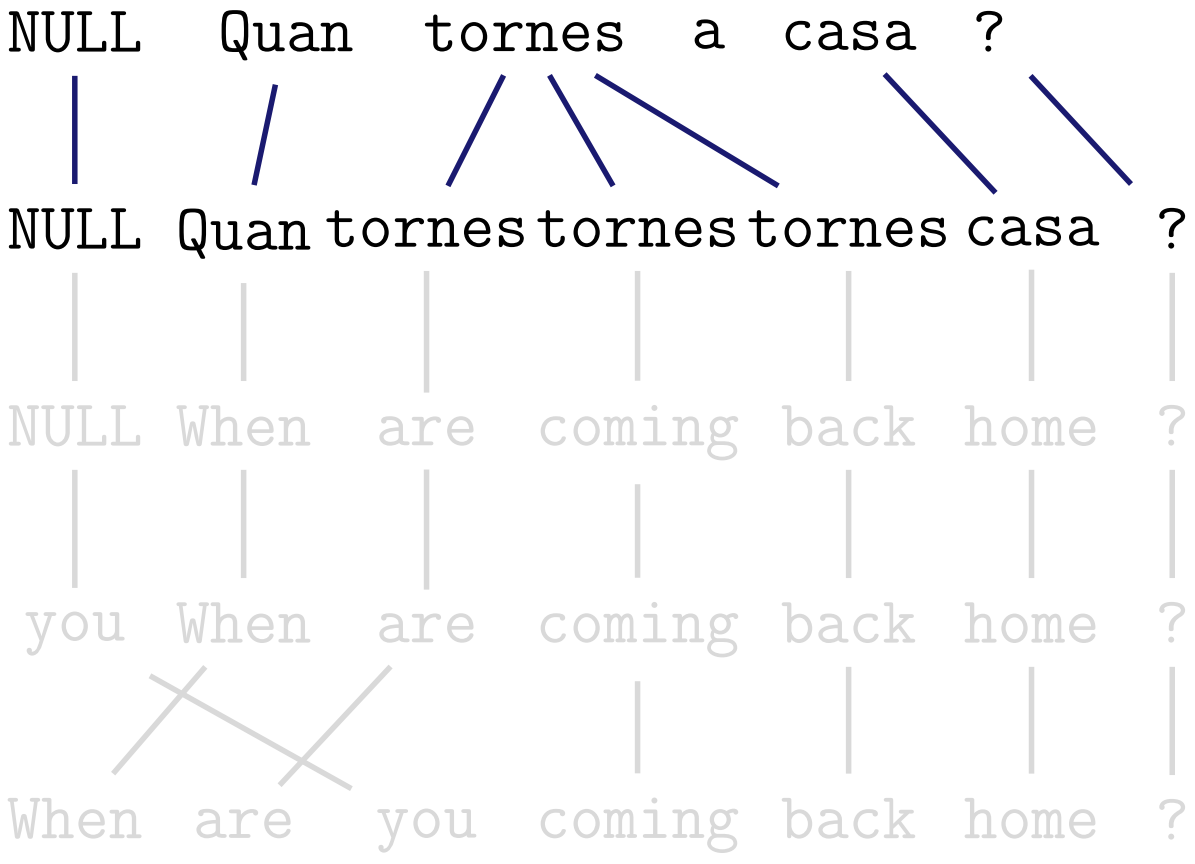
Back to the example:



SMT, components

The translation model $P(f|e)$

Back to the example:



Fertility

Translation

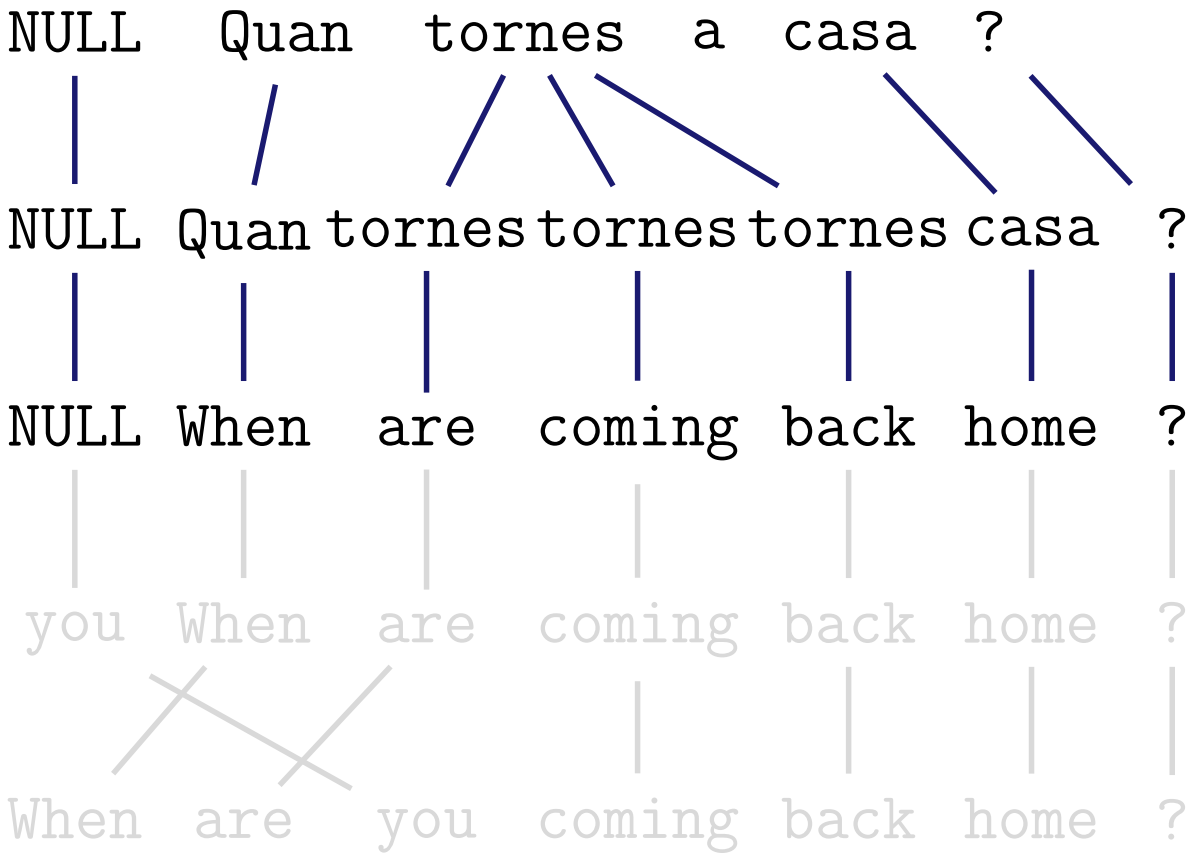
Insertion

Distortion

SMT, components

The translation model $P(f|e)$

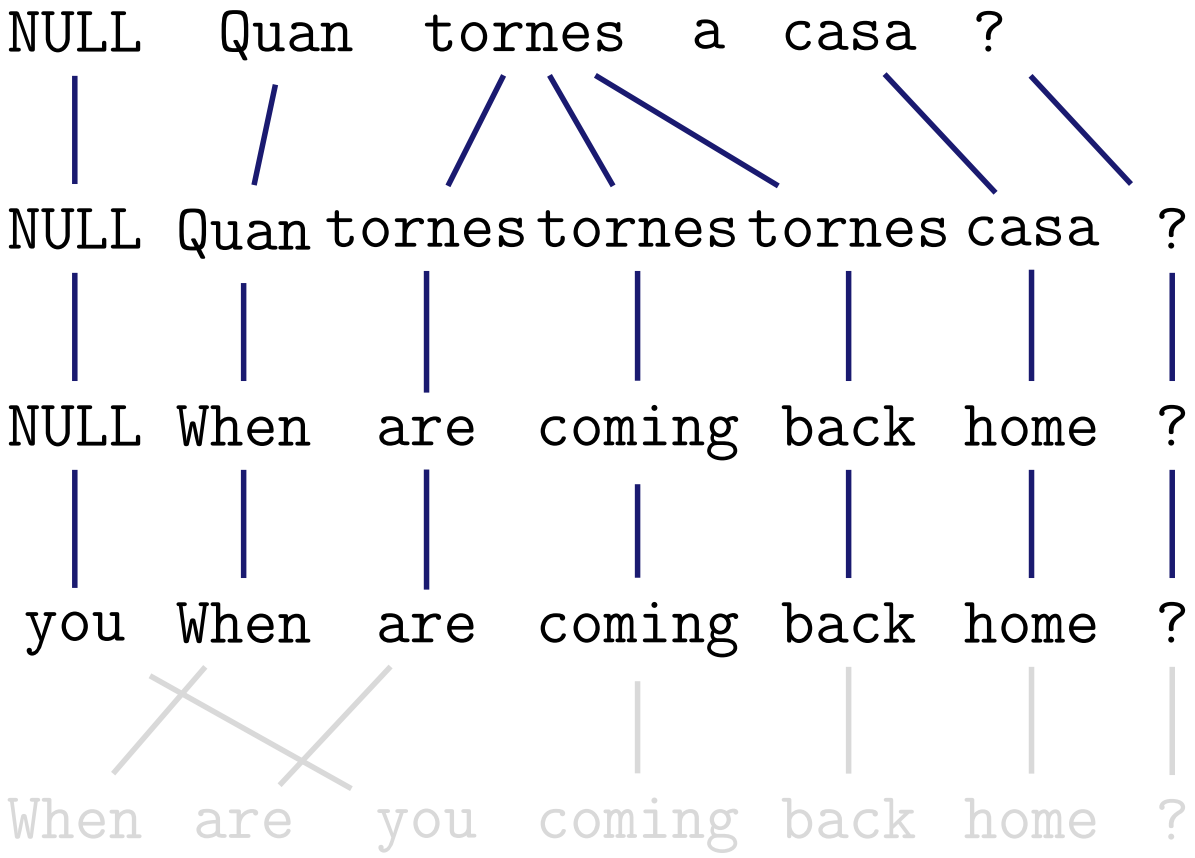
Back to the example:



SMT, components

The translation model $P(f|e)$

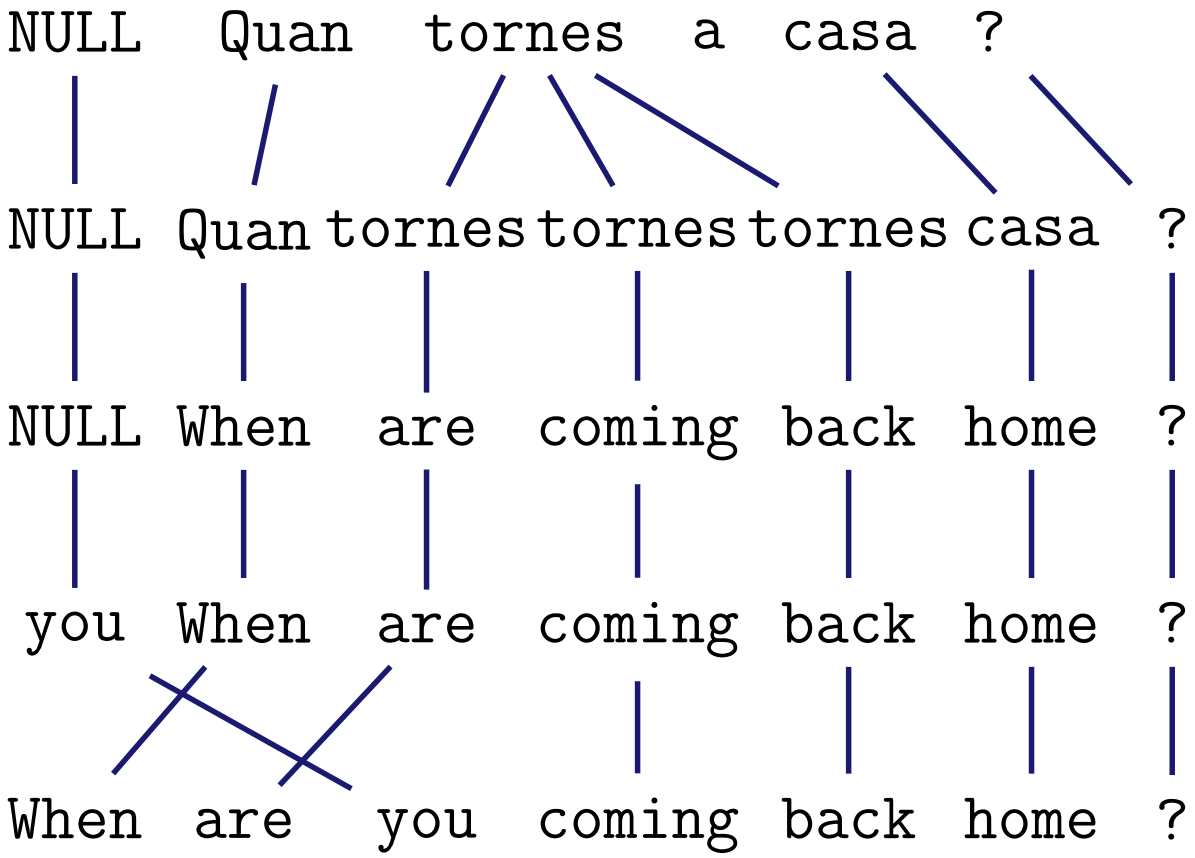
Back to the example:



SMT, components

The translation model $P(f|e)$

Back to the example:



IBM 1 (Philipp Koehn's slide)

- Generative model: break up translation into smaller steps
  - IBM Model 1 only uses lexical translation
- Translation probability
  - for a foreign sentence $\mathbf{f} = (f_1, ..., f_{l_f})$ of length l_f
  - to an English sentence $\mathbf{e} = (e_1, ..., e_{l_e})$ of length l_e
  - with an alignment of each English word e_j to a foreign word f_i according to the alignment function $a : j \rightarrow i$

$$p(\mathbf{e}, a|\mathbf{f}) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})$$

parameter ϵ is a normalization constant

SMT, components

The translation model $P(f|e)$

IBM 1 (Example from Philipp Koehn's slides)

das

e	$t(e\|f)$
the	0.7
that	0.15
which	0.075
who	0.05
this	0.025

Haus

e	$t(e\|f)$
house	0.8
building	0.16
home	0.02
household	0.015
shell	0.005

ist

e	$t(e\|f)$
is	0.8
's	0.16
exists	0.02
has	0.015
are	0.005

klein

e	$t(e\|f)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

$$p(e,a|f) = \frac{\epsilon}{4^3} \times t(\textbf{the}|\textbf{das}) \times t(\textbf{house}|\textbf{Haus}) \times t(\textbf{is}|\textbf{ist}) \times t(\textbf{small}|\textbf{klein})$$

$$= \frac{\epsilon}{4^3} \times 0.7 \times 0.8 \times 0.8 \times 0.4 = 0.0028\epsilon$$

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

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

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!

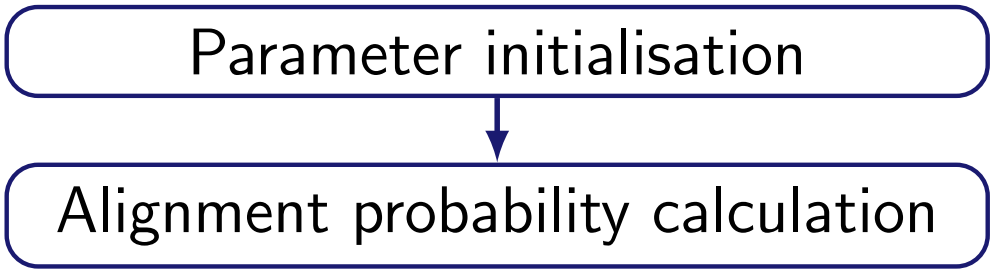
But...

- 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

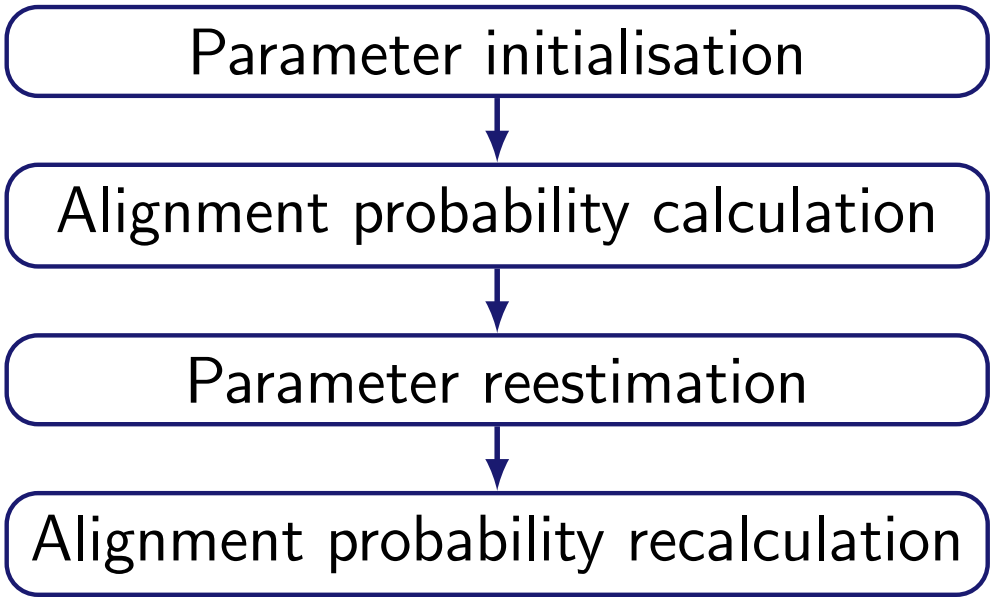
Expectation-Maximisation algorithm



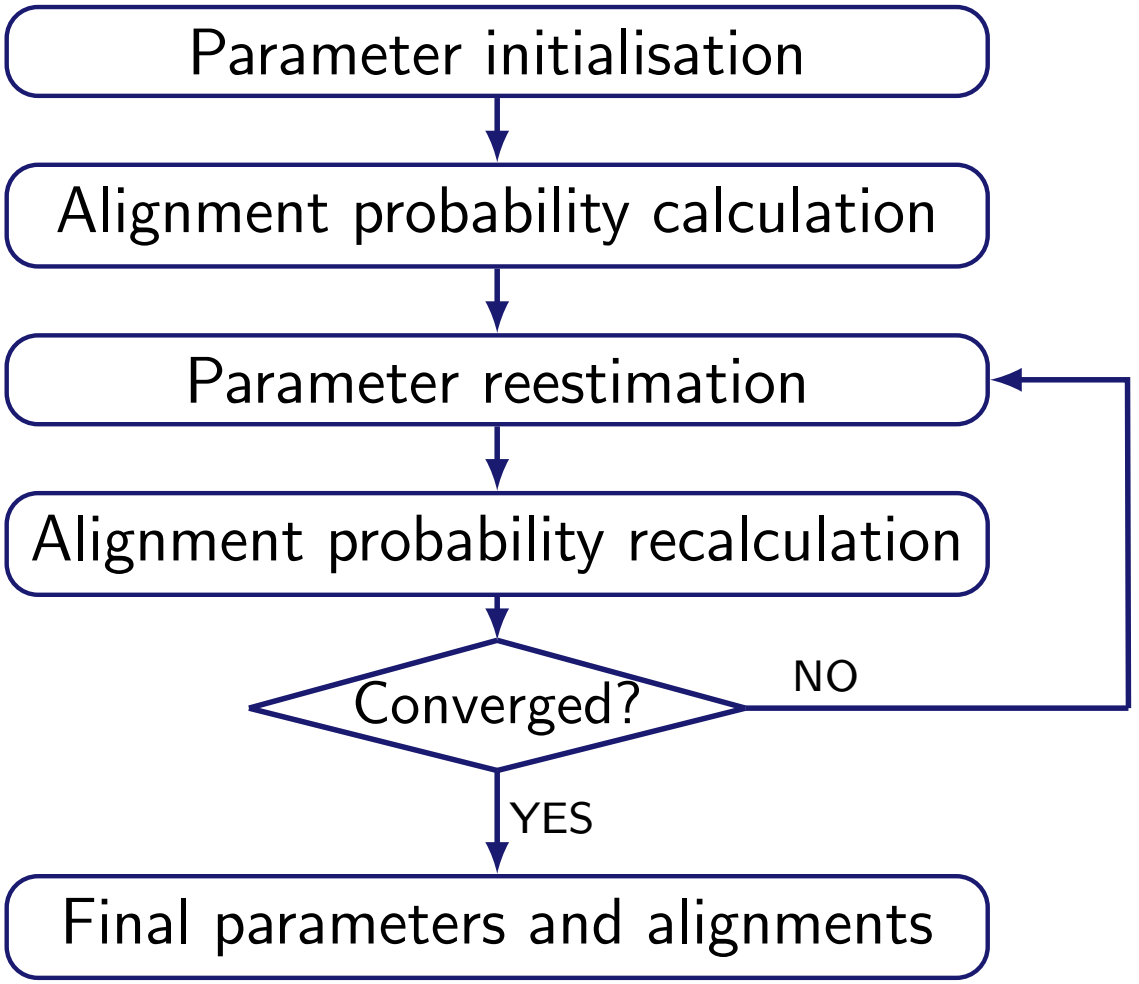
SMT, components

The translation model $P(f|e)$

Expectation-Maximisation algorithm



Expectation-Maximisation algorithm



SMT, components

The translation model $P(f|e)$

IBM Model 1 lexical translation: t

IBM Model 2 adds absolute reordering model: t, d

IBM Model 3 adds fertility model: t, d, n

IBM Model 4 relative reordering model: t, d, n

IBM Model 5 fixes deficiency by keeping track of vacancies: t, d, n

IBM Models

- Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. *A statistical approach to machine translation.* Computational Linguistics, 16(2):79-85, 1990

- Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2004

  `http://www.statmt.org/book/slides/04-word-based-models.pdf`

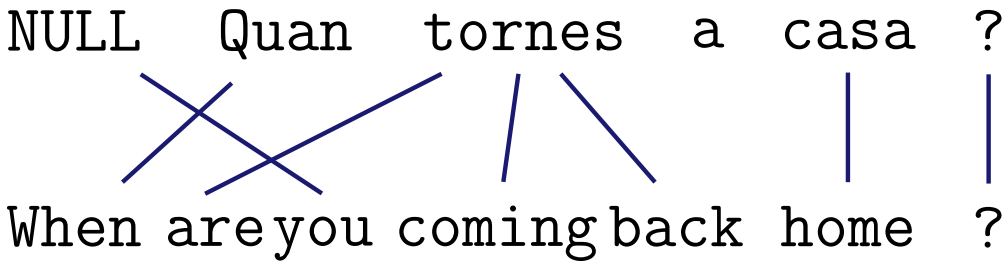
Wait!

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



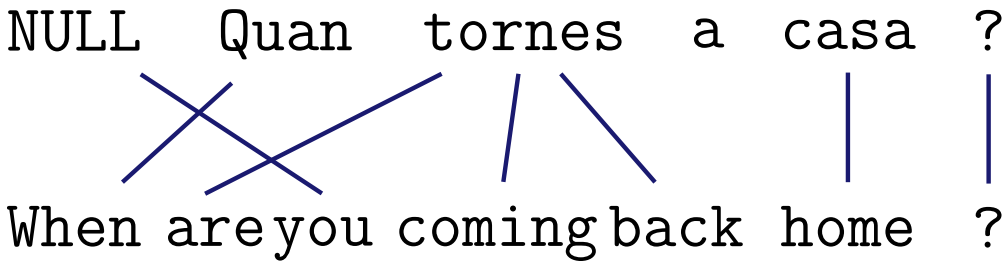
English to Catalan

Alignment's asymmetry

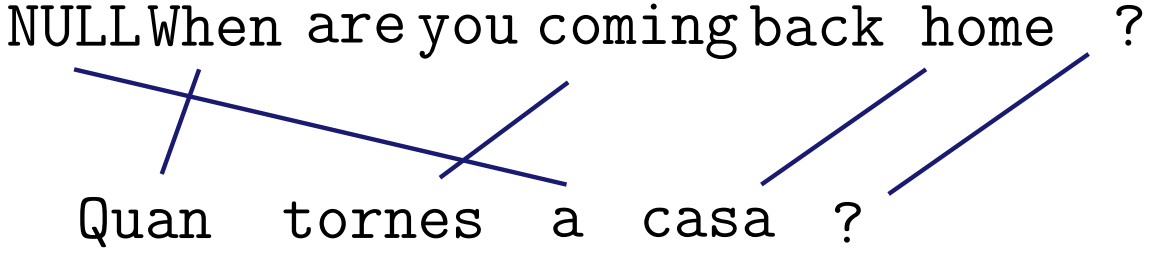
The definitions in IBM models make the alignments asymmetric

- 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 $P(f|e)$

Visually:

Catalan to English

SMT, components

The translation model $P(f|e)$

Visually:

English to Catalan

SMT, components

The translation model $P(f|e)$

Alignment symmetrisation

- Intersection: high-confidence, high precision.

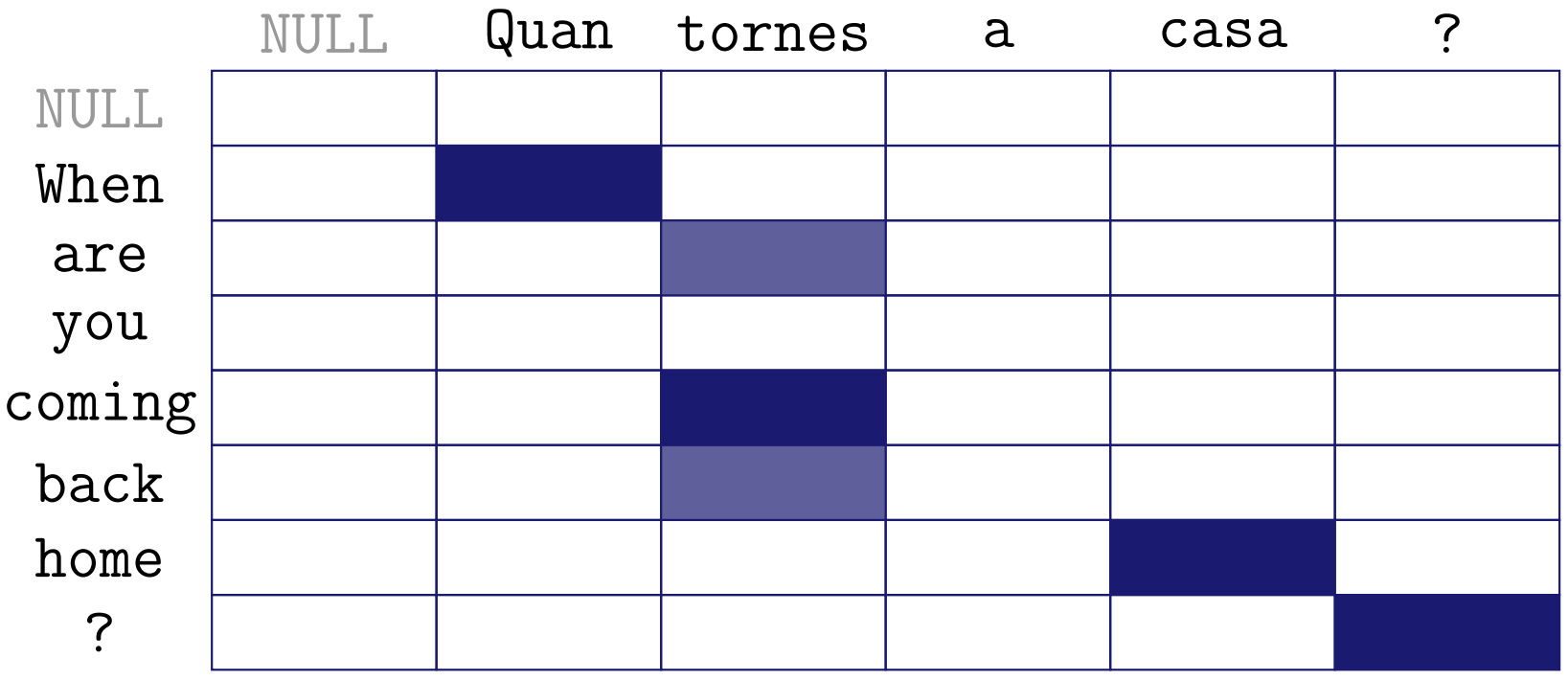
Catalan to English $\bigcap$ English to Catalan

SMT, components

The translation model $P(f|e)$

Alignment symmetrisation

- Union: lower confidence, high recall.



Catalan to English $\bigcup$ English to Catalan

SMT, components

The translation model $P(f|e)$

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 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. ✓

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

SMT, components

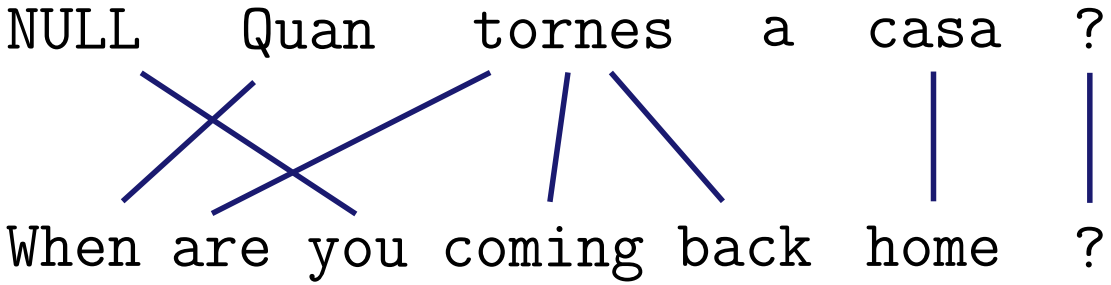
The translation model $P(f|e)$

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 $P(f|e)$

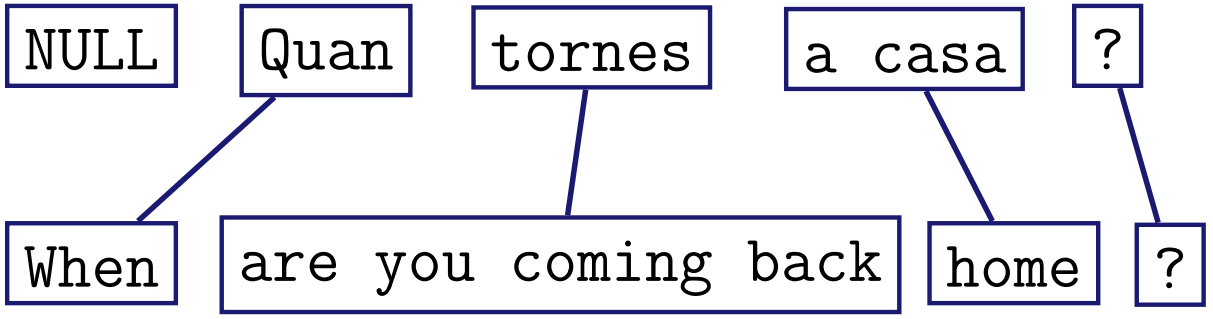


With the new translation units, $P(f|e)$ can be obtained following the same strategy as for word-based models with few modifications:

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

SMT, components

The translation model $P(f|e)$

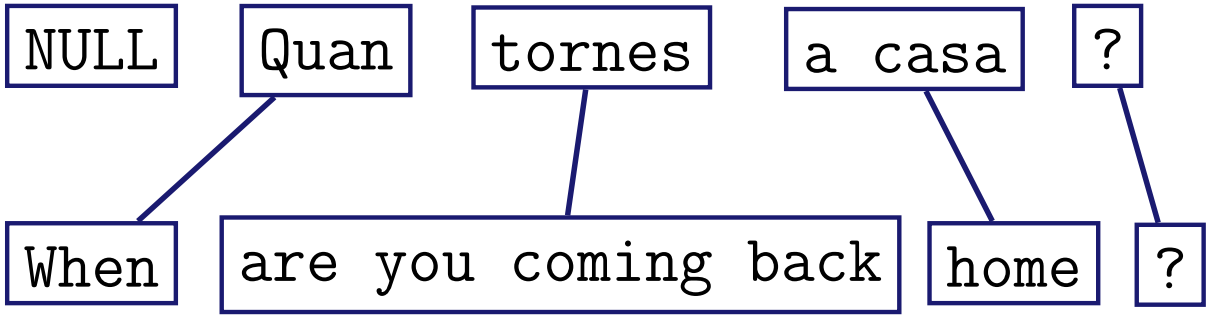


With the new translation units, $P(f|e)$ can be obtained following the same strategy as for word-based models with few modifications:

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

SMT, components

The translation model $P(f|e)$

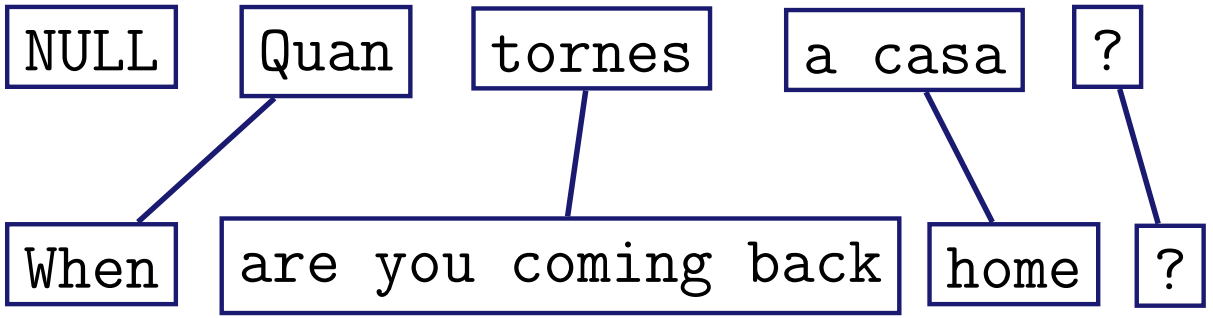


With the new translation units, $P(f|e)$ can be obtained following the same strategy as for word-based models with few modifications:

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.

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]

[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:

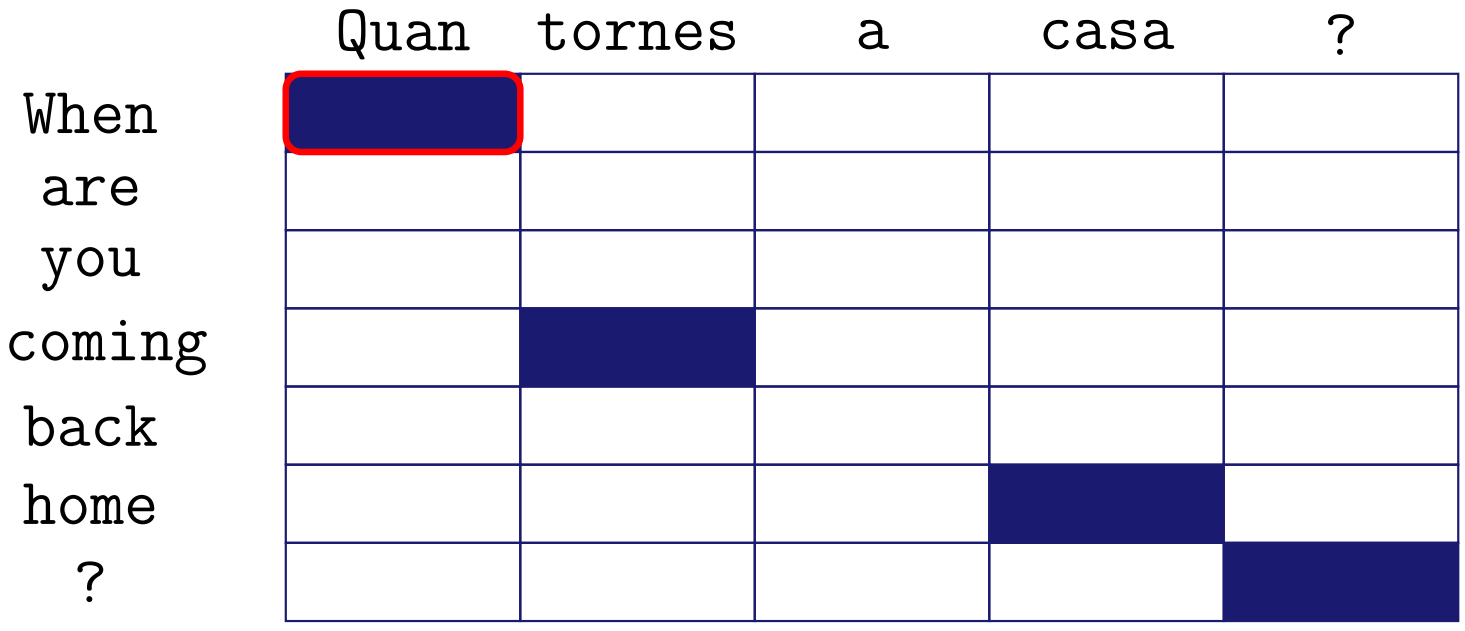
~~(Quan tornes, When are you coming back)~~

(Quan tornes tu, When are you coming back)

SMT, components

The translation model $P(f|e)$

Intersection

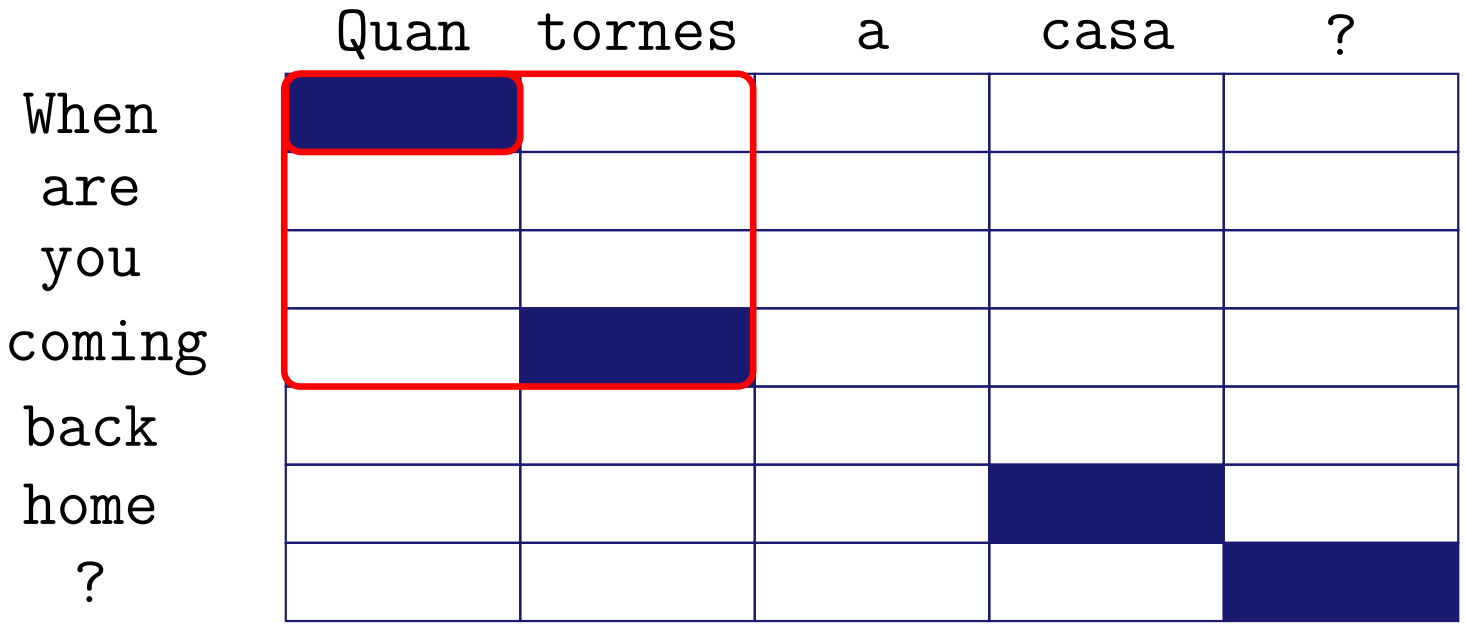


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

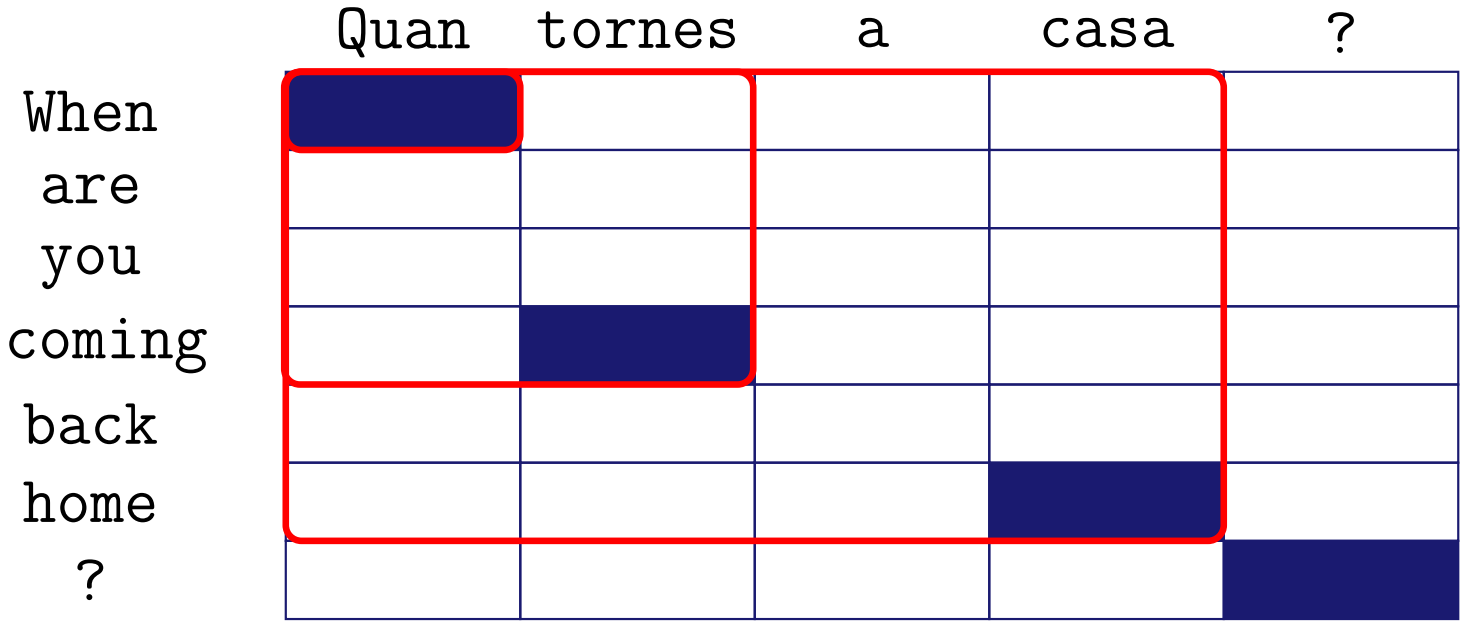


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

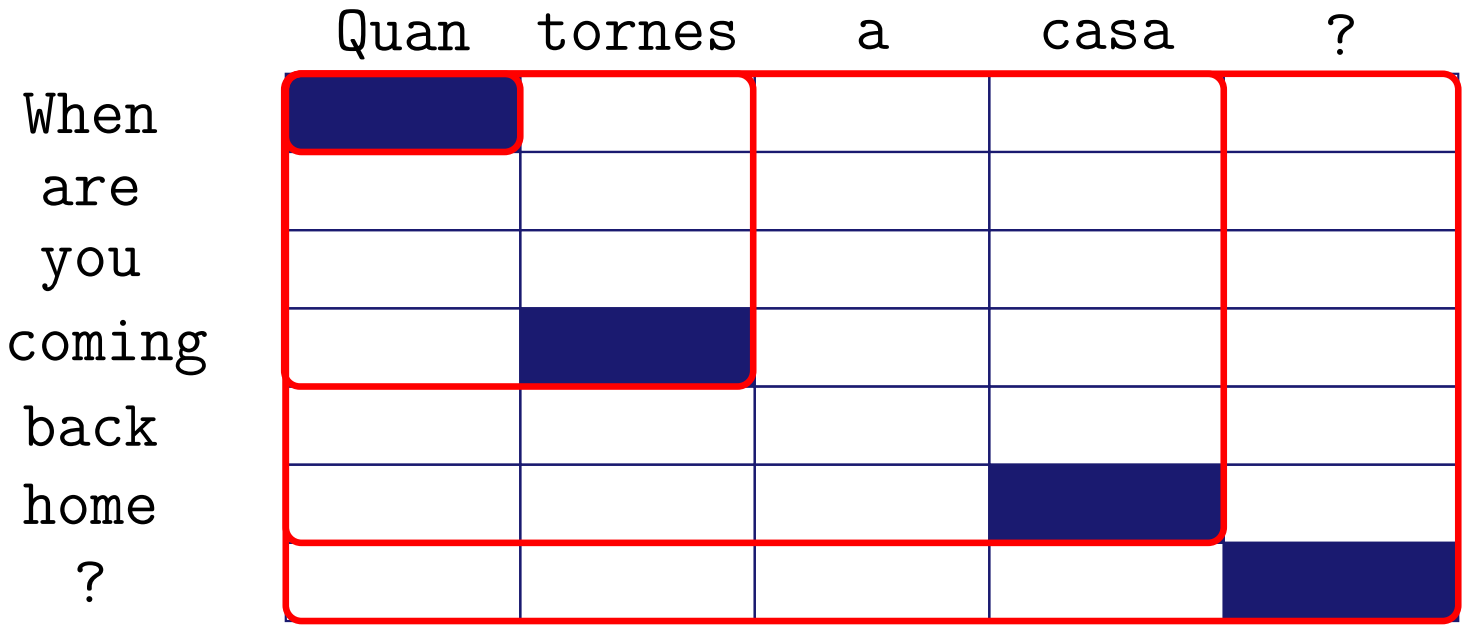


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

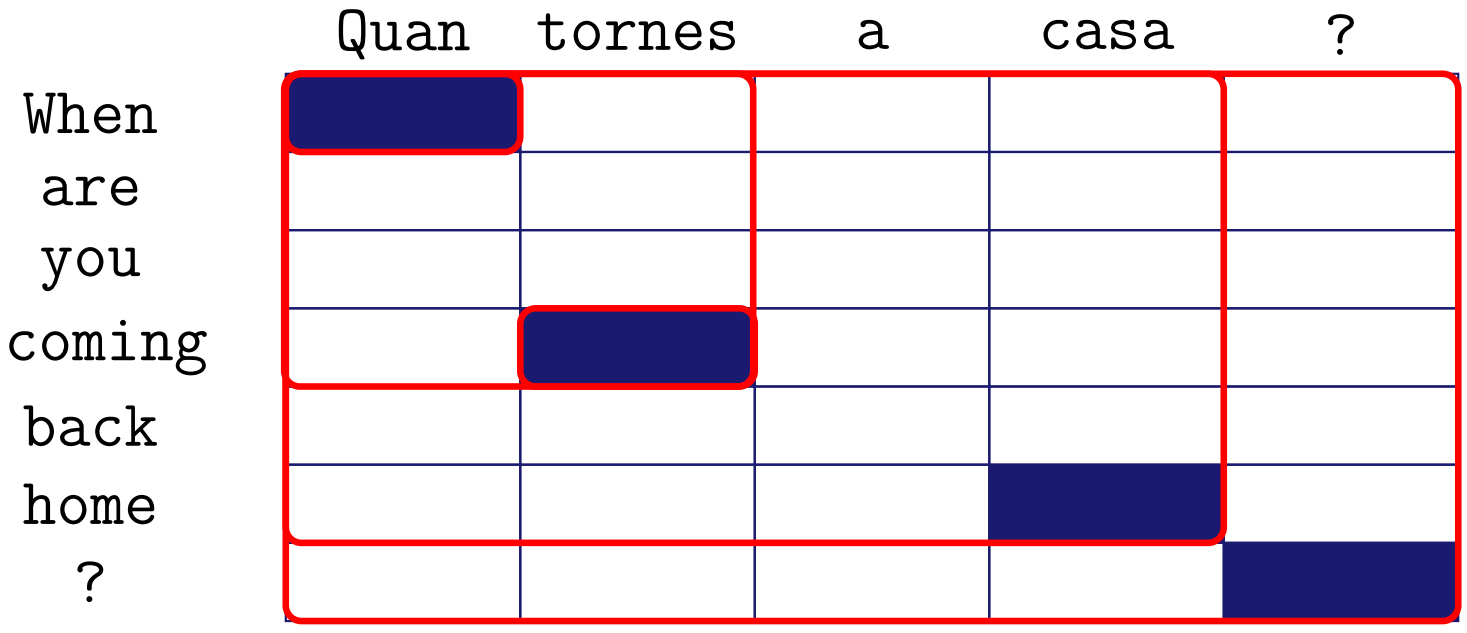


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

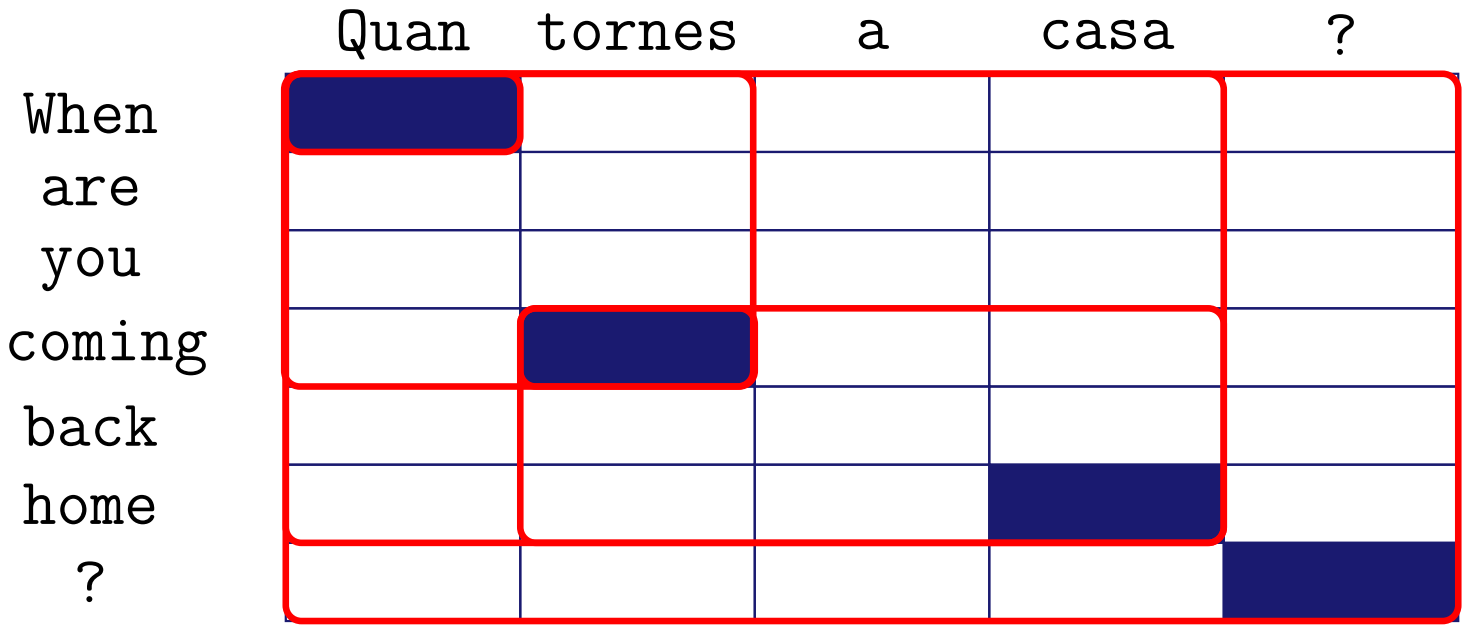


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

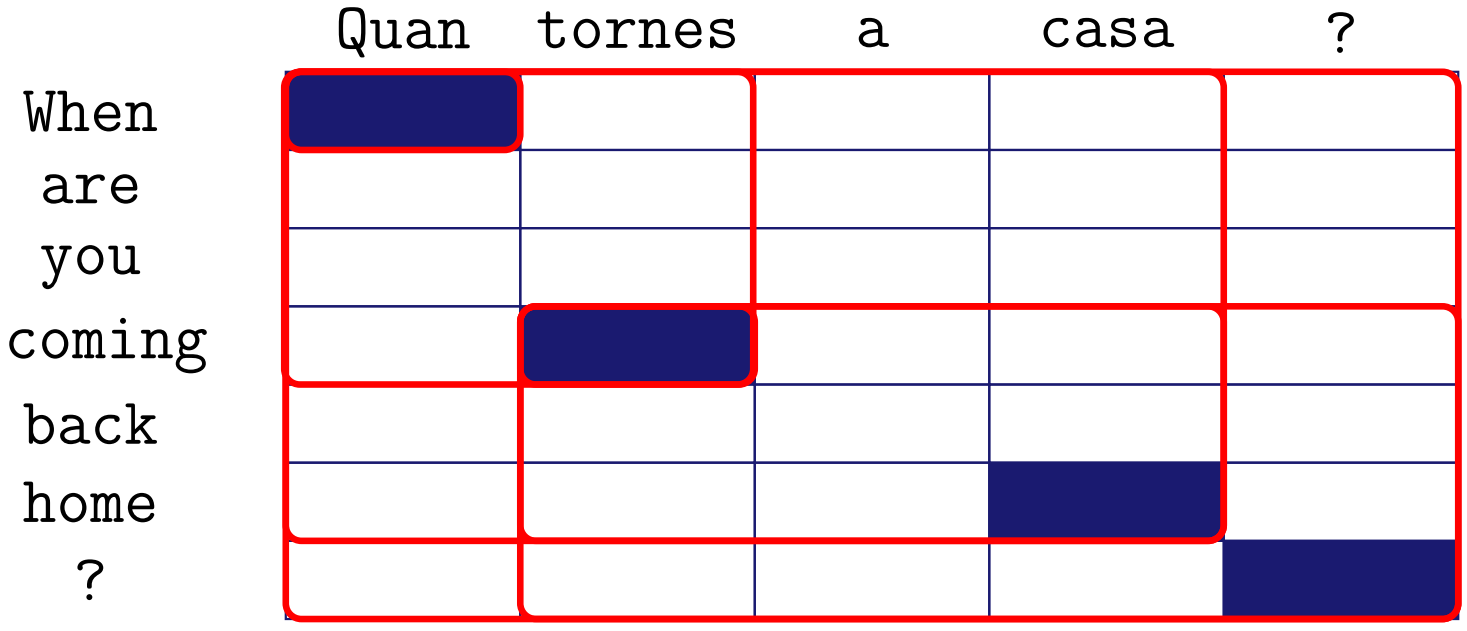


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

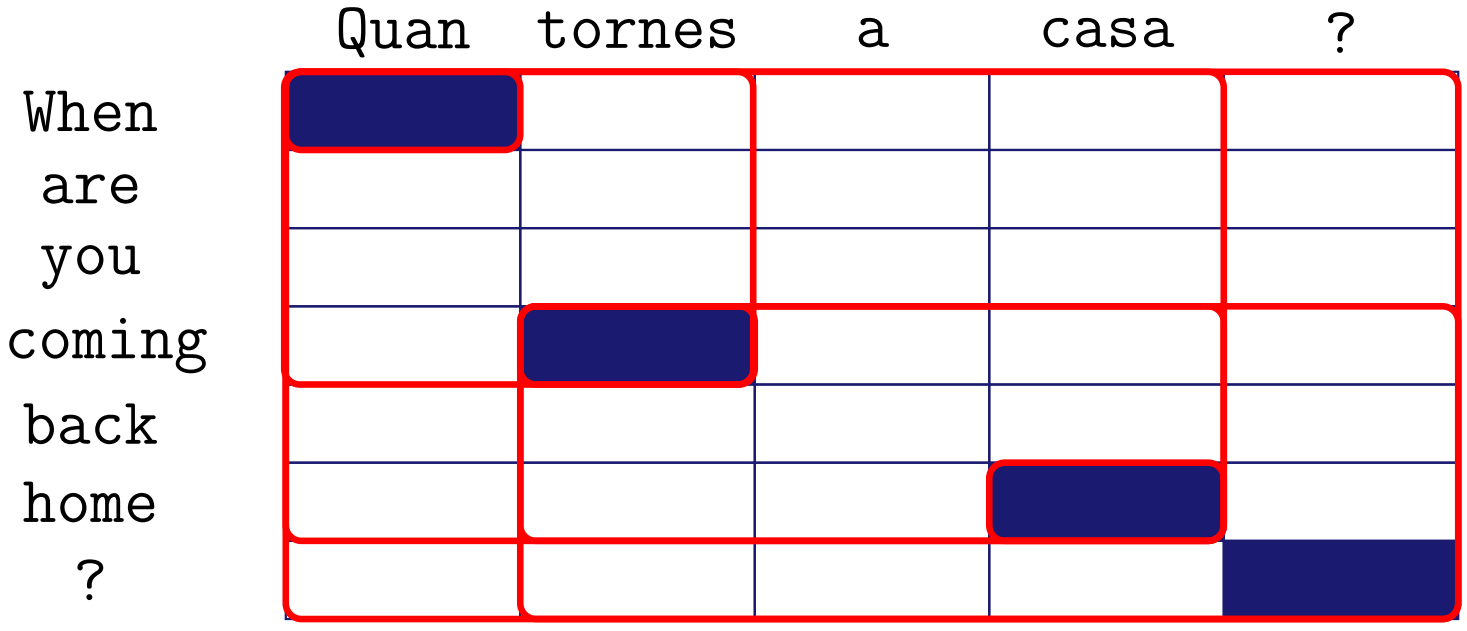


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

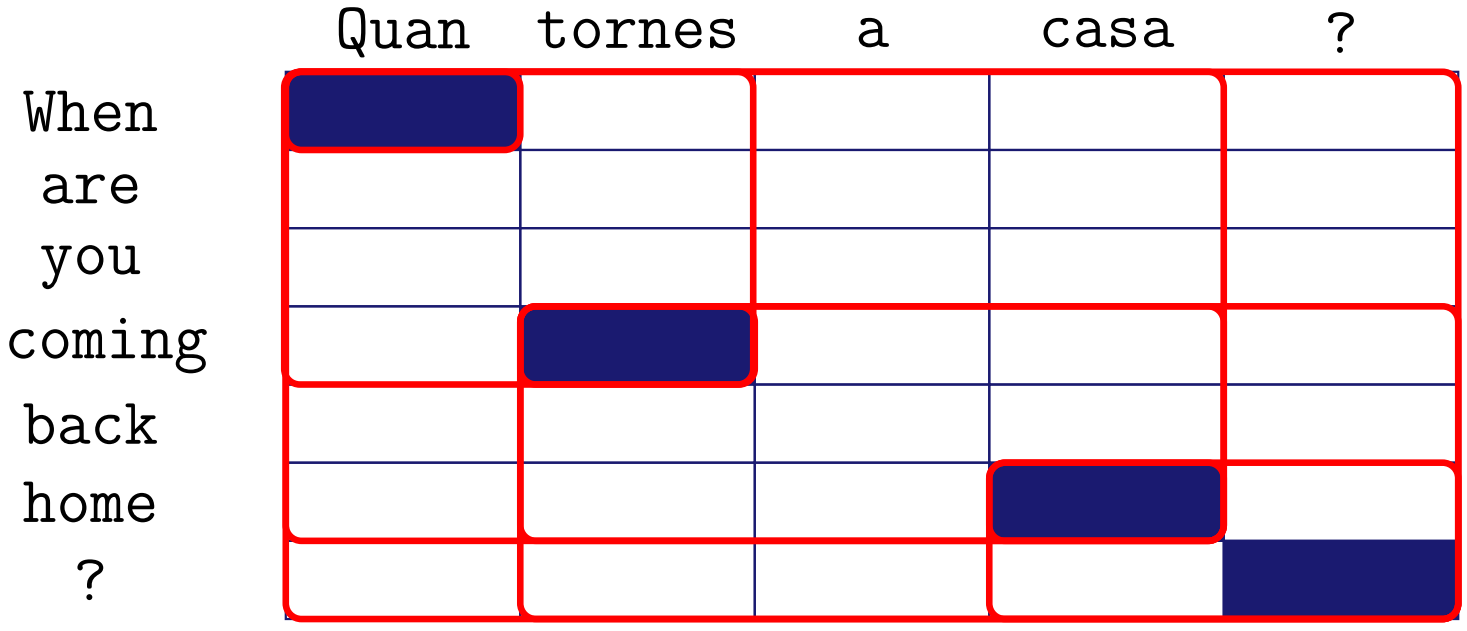


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

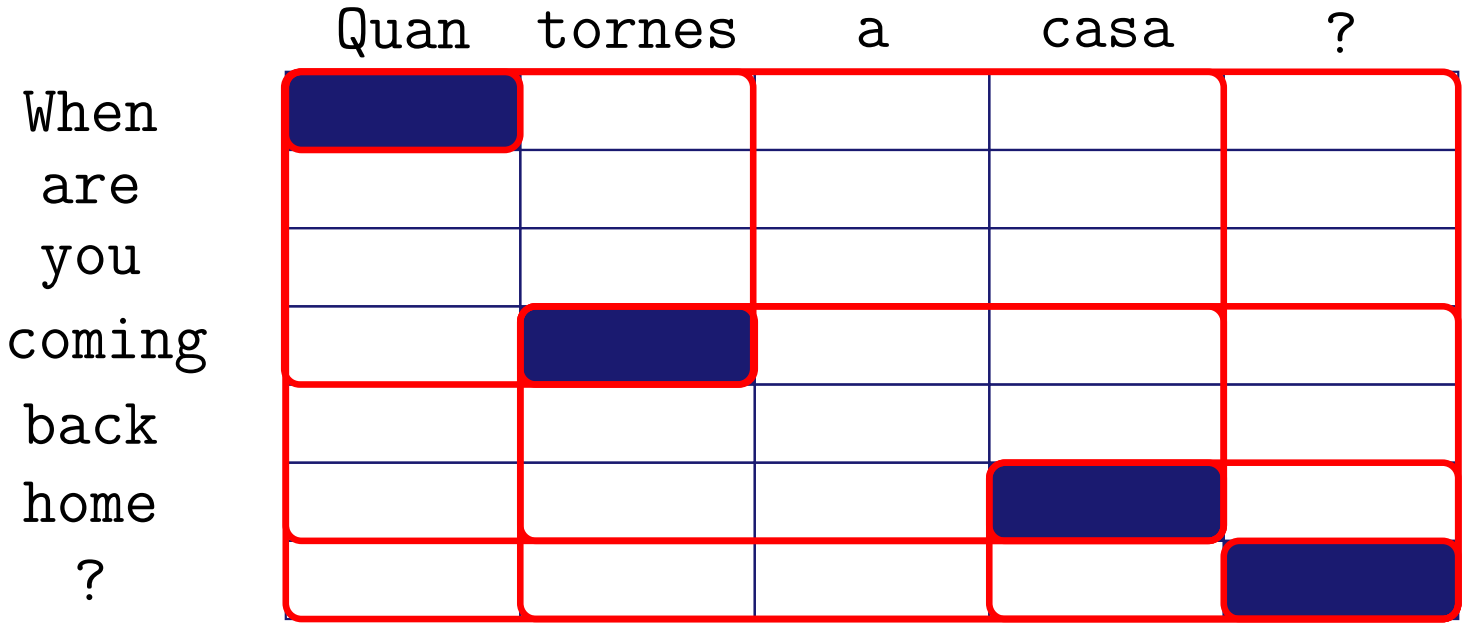


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

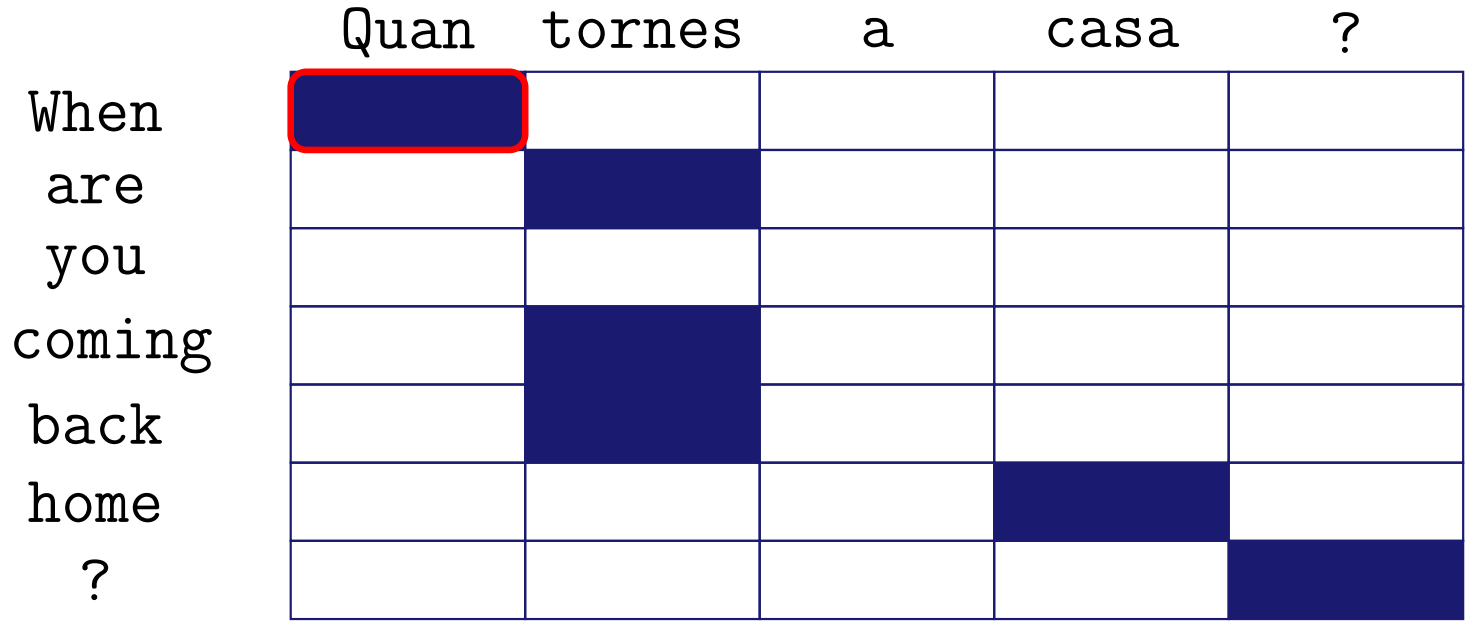


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Union

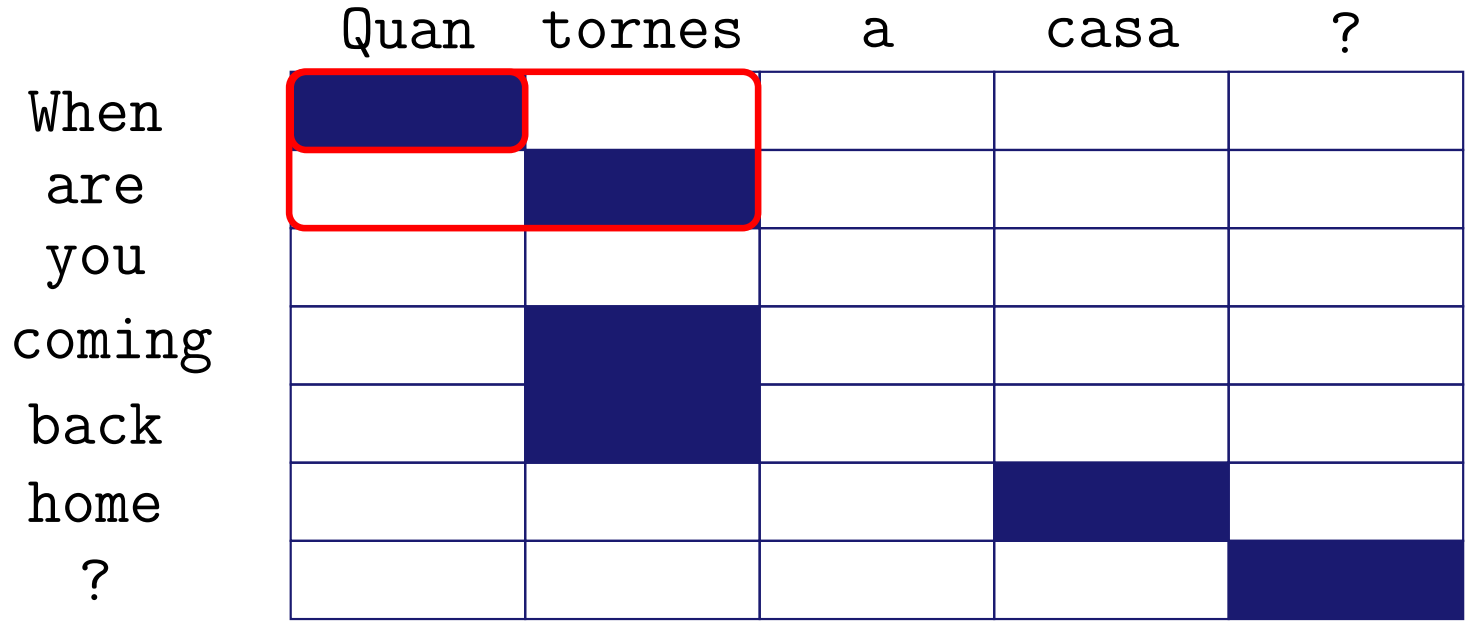


(Quan, When) (Quan tornes, When are) (Quan tornes, When are you coming) (Quan tornes, When are you coming back) (Quan tornes a casa, When are you coming back home) ... (tornes a casa ?, are you coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 21 phrases

SMT, components

The translation model $P(f|e)$

Union

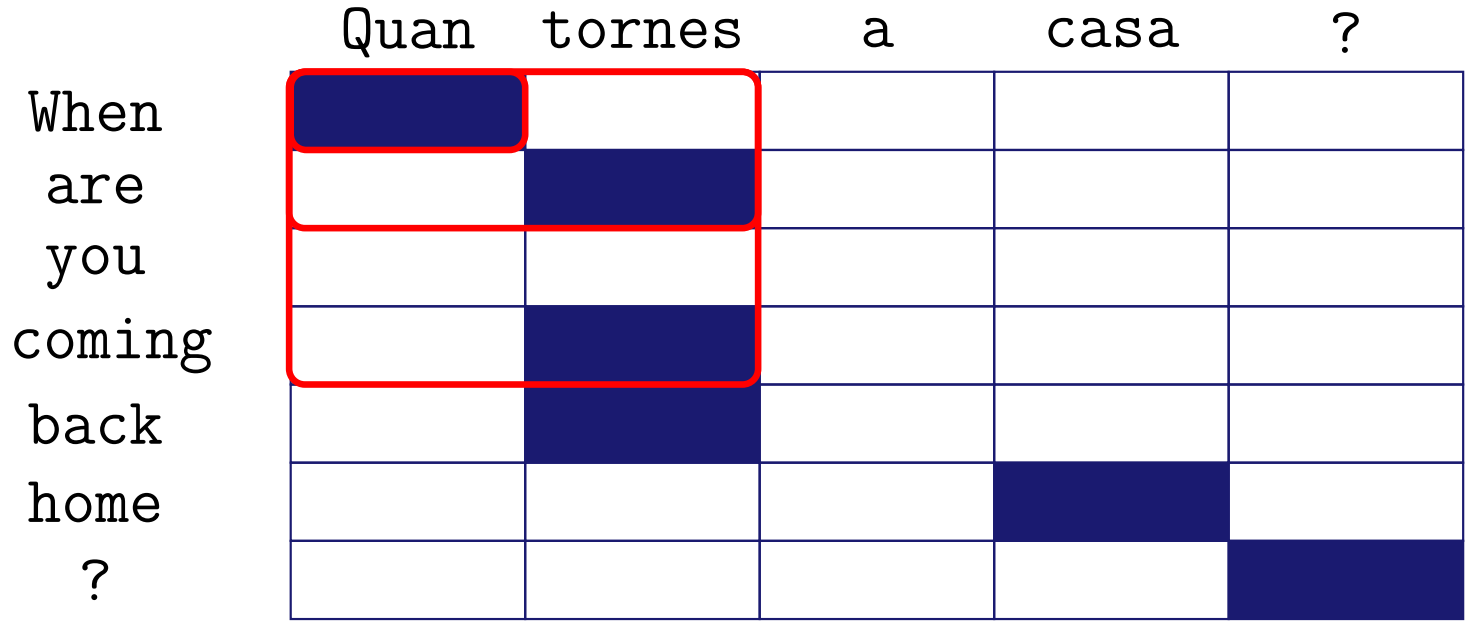


(Quan, When) (Quan tornes, When are) (Quan tornes, When are you coming) (Quan tornes, When are you coming back) (Quan tornes a casa, When are you coming back home) ... (tornes a casa ?, are you coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 21 phrases

SMT, components

The translation model $P(f|e)$

Union

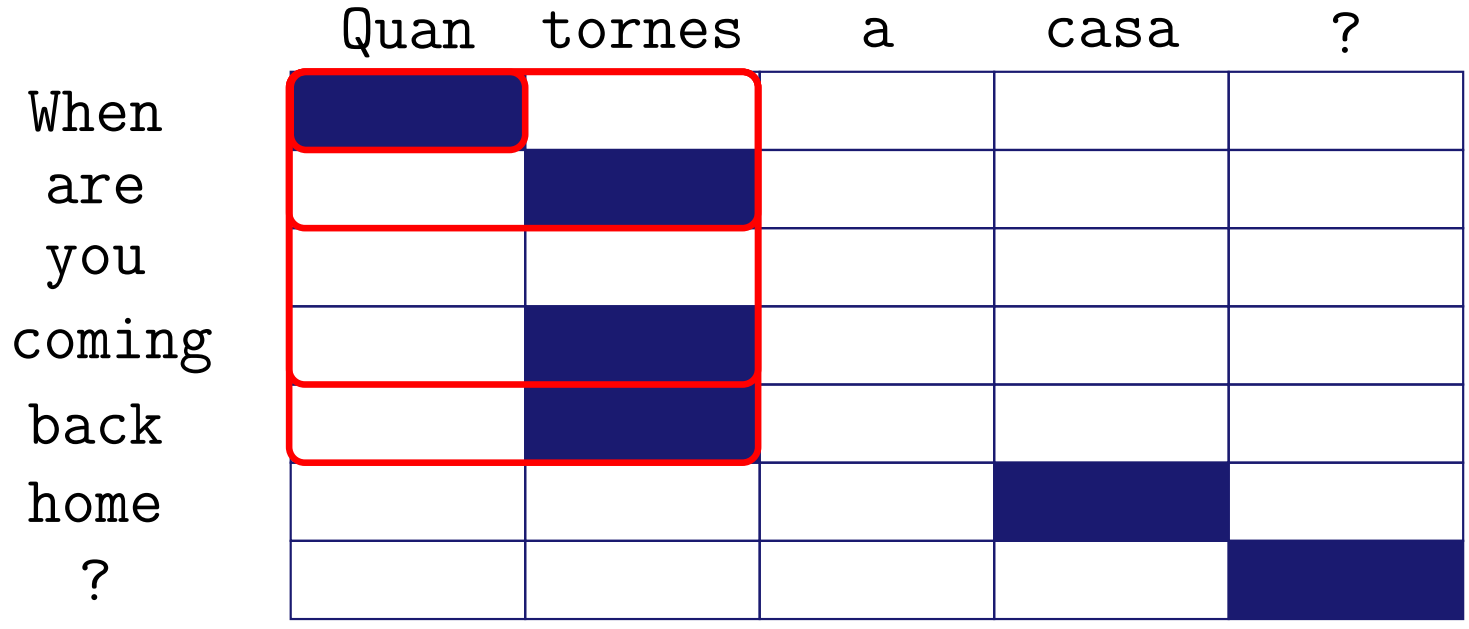


(Quan, When) (Quan tornes, When are) (Quan tornes, When are you coming) (Quan tornes, When are you coming back) (Quan tornes a casa, When are you coming back home) ... (tornes a casa ?, are you coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 21 phrases

SMT, components

The translation model $P(f|e)$

Union

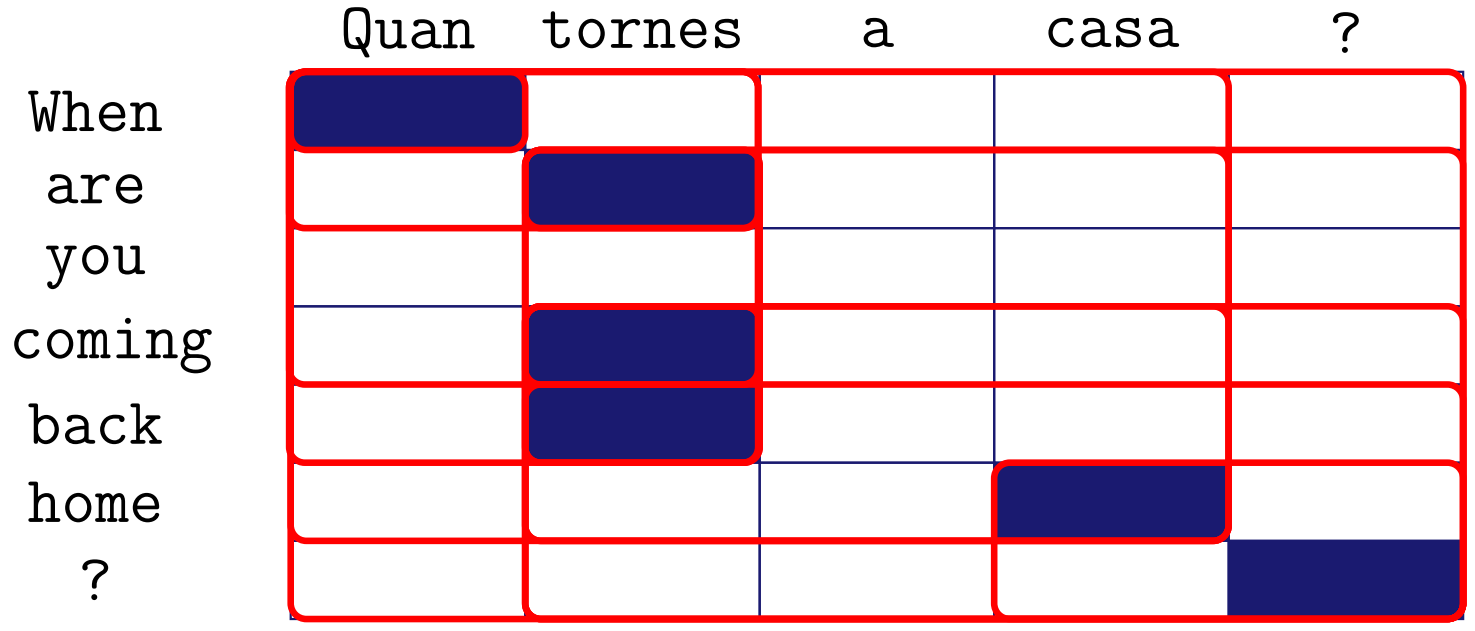


(Quan, When) (Quan tornes, When are) (Quan tornes, When are you coming) (Quan tornes, When are you coming back) (Quan tornes a casa, When are you coming back home) ... (tornes a casa ?, are you coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 21 phrases

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

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.

SMT, components

The translation model $P(f|e)$

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
...
```

SMT, components

The translation model $P(f|e)$

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.

Wait!

RECAP!

SMT is Simple...

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

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*

RECAP!

SMT is Simple...

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

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, components

Decoder

Decoder

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

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
- Beam search decoders
- Others (A* search...)

SMT, components

Decoder

Decoder

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

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
- Beam search decoders
- Others (A* search...)

SMT, components

Decoder

Note!

These algorithms are not limited to SMT,
they are general for decoding/generation

Greedy decoders

Translate word by word taking always the most probable next token

- Refinement: Initial hypothesis (word by word translation) refined iteratively using hill-climbing heuristics.
- Drawback: It might get trapped in loops where the same output is repeated forever

Greedy Hill-Climbing (Koehn's book slide)

- Create one complete hypothesis with depth-first search (or other means)
- Search for better hypotheses by applying change operators
  - change the translation of a word or phrase
  - combine the translation of two words into a phrase
  - split up the translation of a phrase into two smaller phrase translations
  - move parts of the output into a different position
  - swap parts of the output with the output at a different part of the sentence
- Terminates if no operator application produces a better translation

Decoder

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

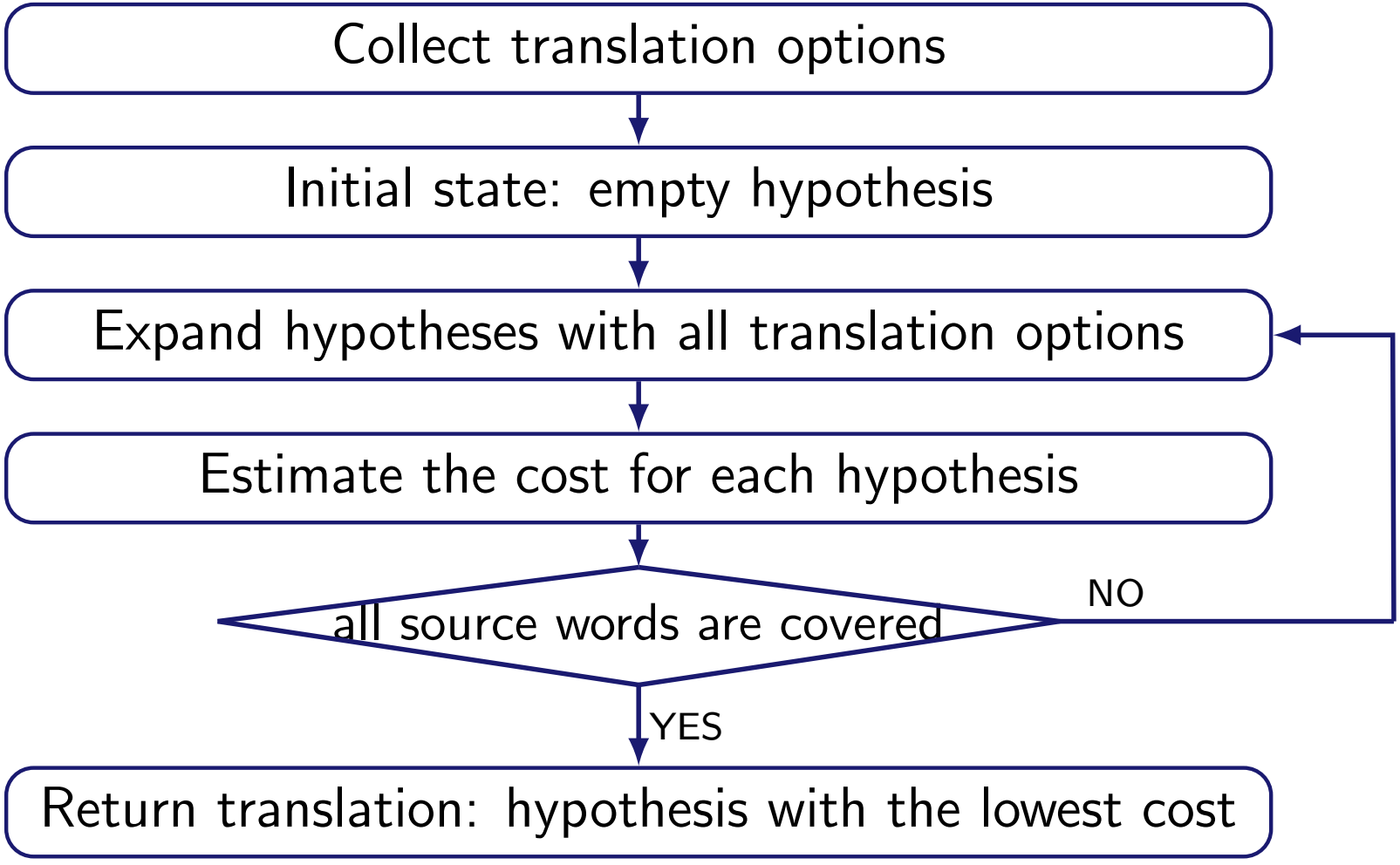
Responsible for the search in the space of possible translations.

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. Let's see now...

Core algorithm



SMT, components

Decoding

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`)

SMT, components

Decoding

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 - -

SMT, components

Decoding

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 - -

SMT, components

Decoding

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

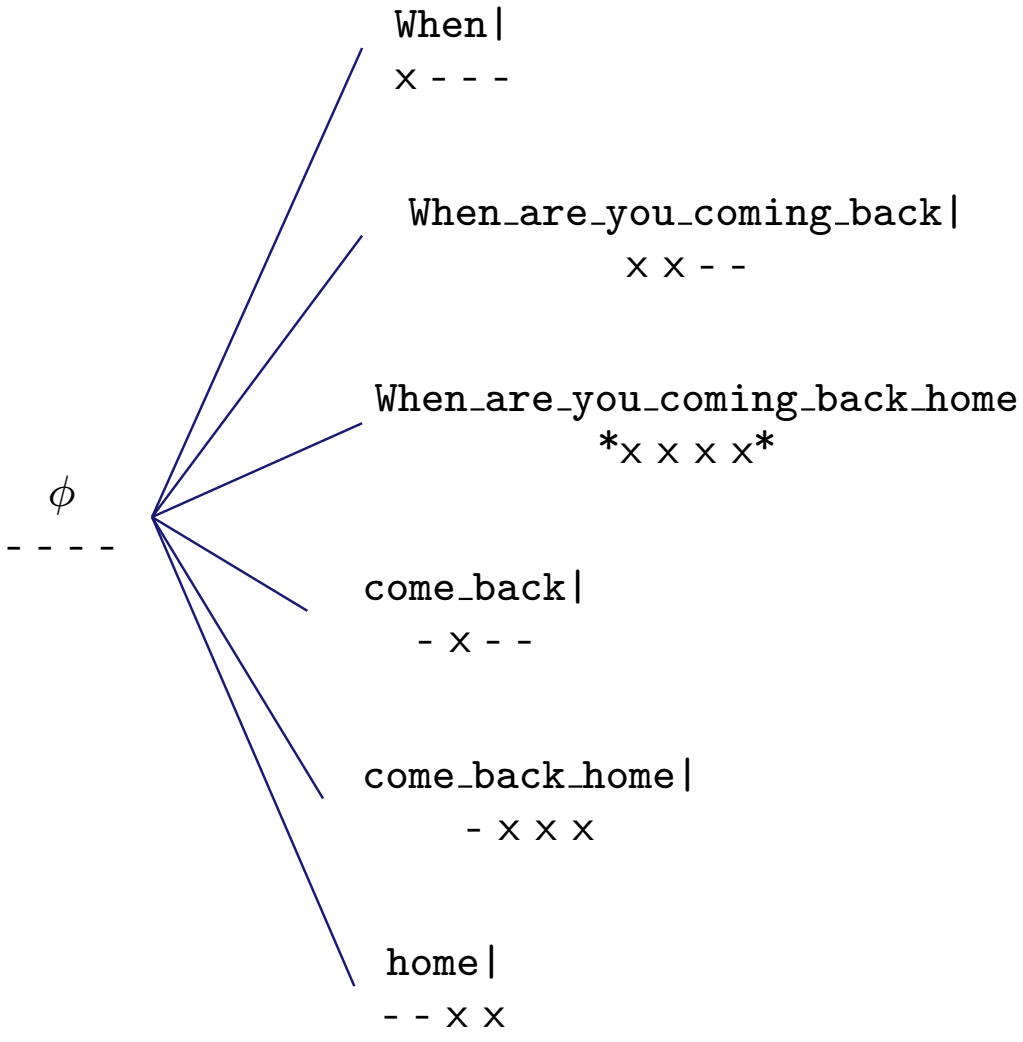
Decoding

$$\phi$$

- - - -

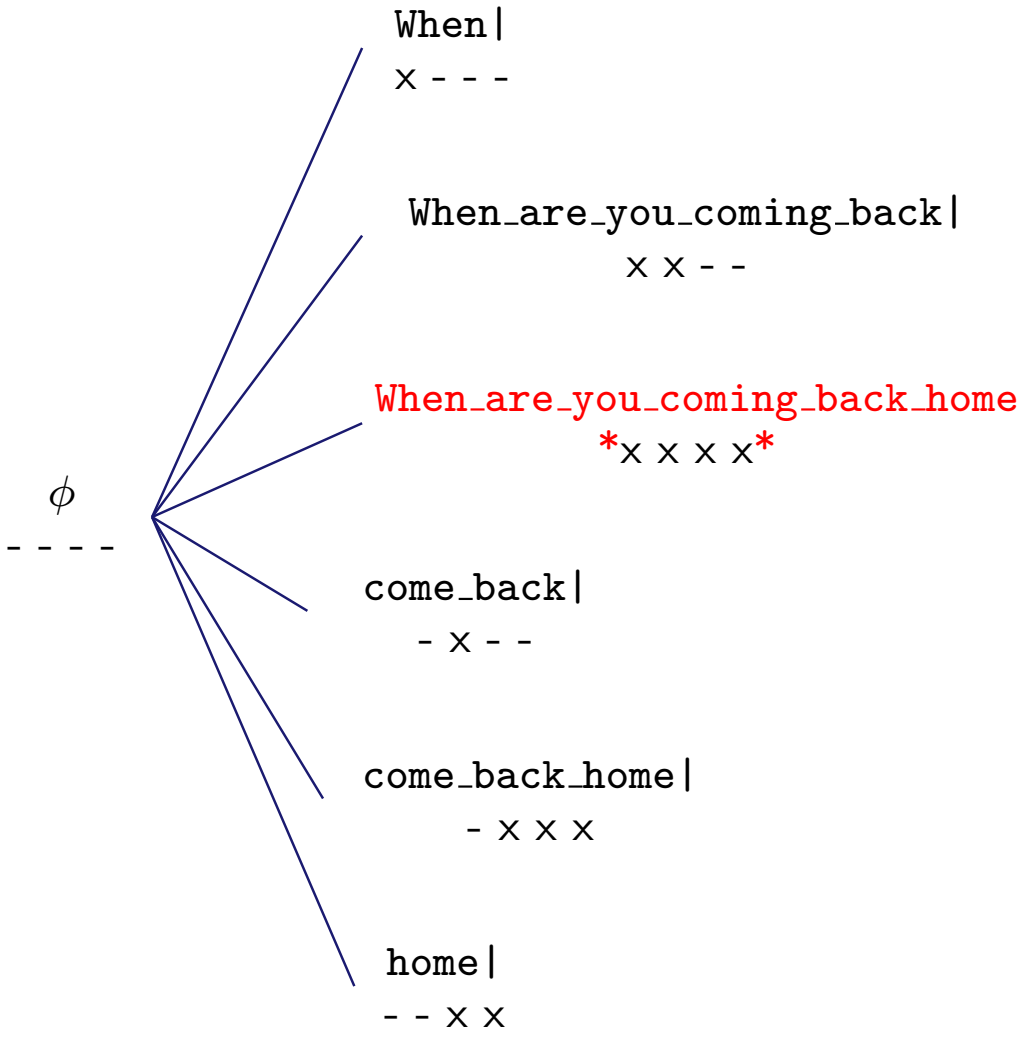
SMT, components

Decoding



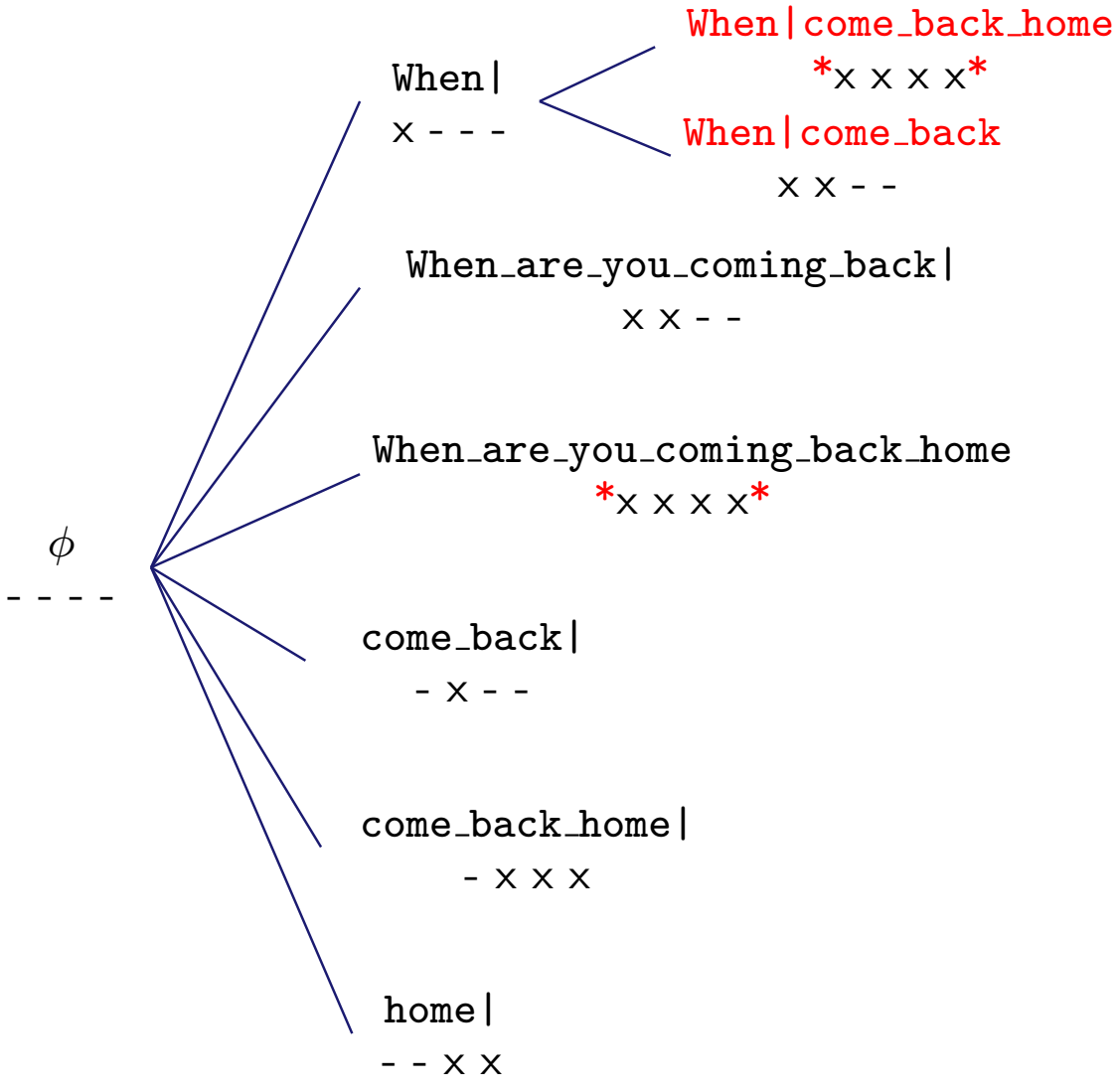
SMT, components

Decoding



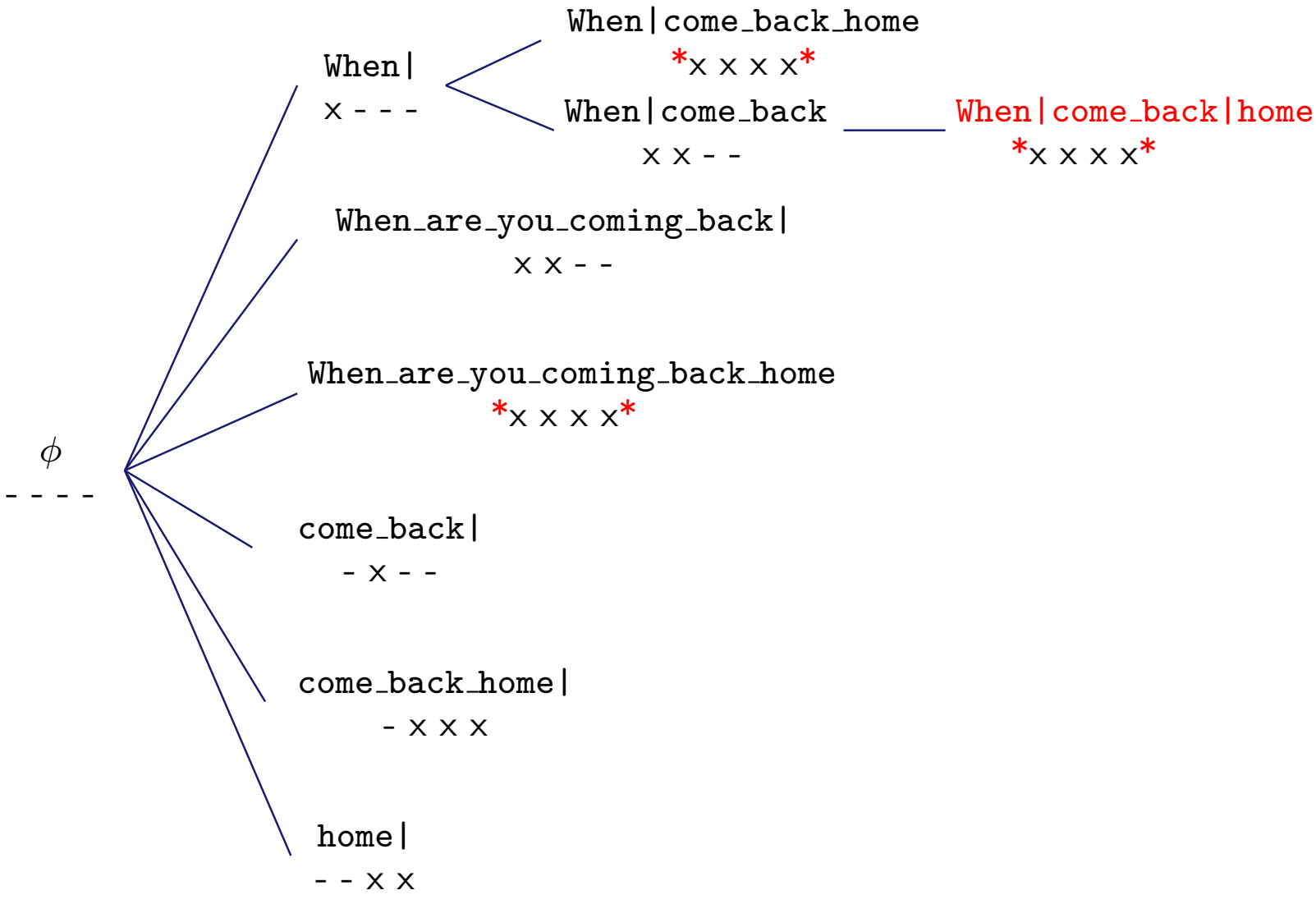
SMT, components

Decoding



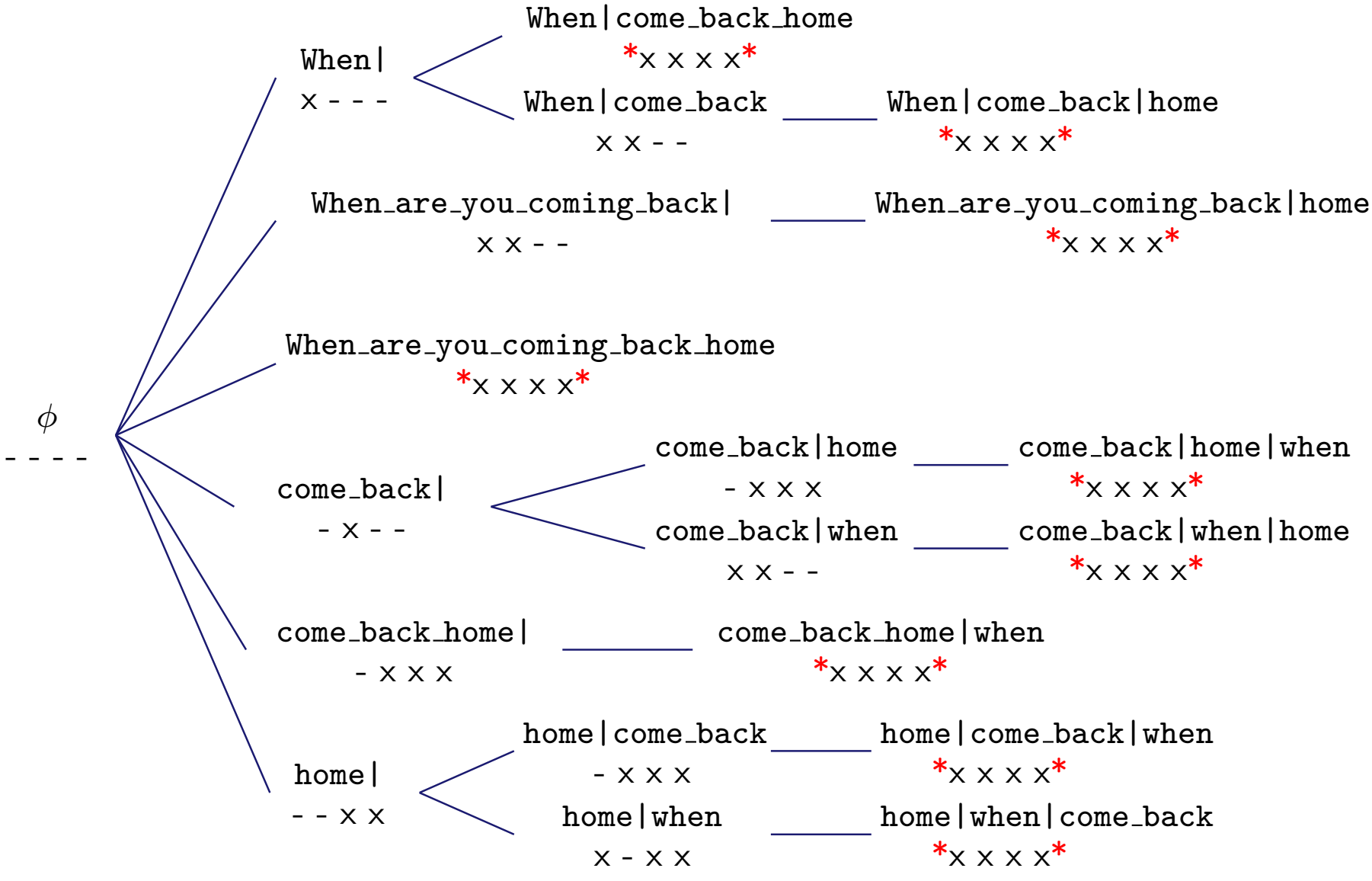
SMT, components

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

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

Hypotheses recombination

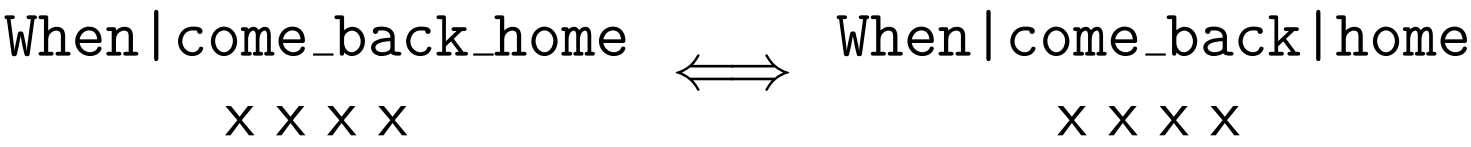
Combine hypotheses with the same source words translated, keep that with a lower cost.

$$\underset{\text{x x x x}}{\texttt{When|come\_back\_home}} \iff \underset{\text{x x x x}}{\texttt{When|come\_back|home}}$$

- 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.



- 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.

$$\begin{array}{c}\texttt{When|come\_back\_home}\\ \text{x x x x}\end{array} \iff \begin{array}{c}\texttt{When|come\_back|home}\\ \text{x x x x}\end{array}$$

- 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!)

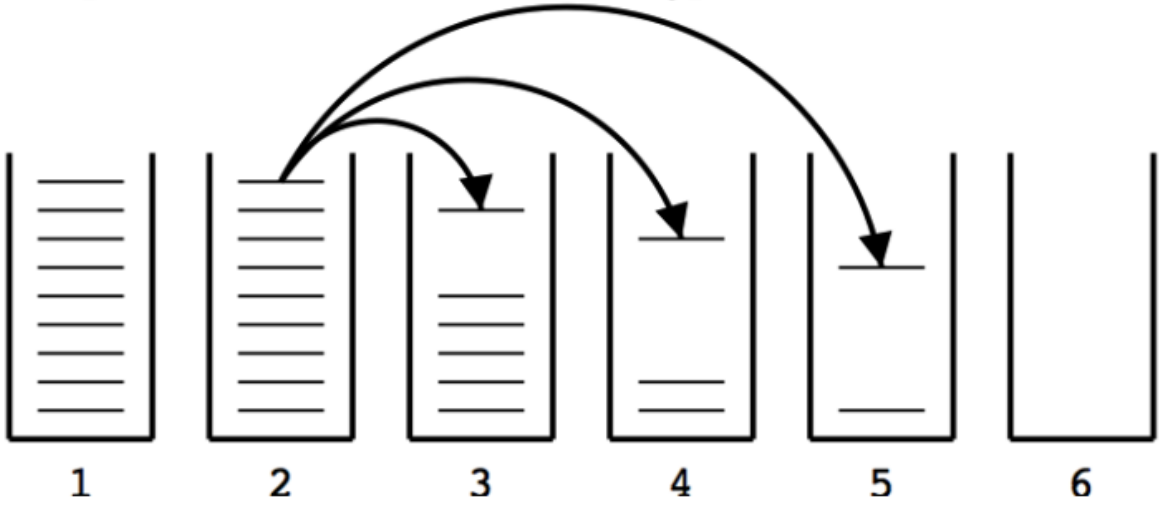
Strategy:

- 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.

SMT, components

Stacks in beam-search decoders (Callison-Burch and Koehn (2005) slides)

```
initialize hypothesisStack[0 .. nf];
create initial hypothesis hyp_init;
add to stack hypothesisStack[0];
for i=0 to nf-1:
  for each hyp in hypothesisStack[i]:
    for each new_hyp that can be derived from hyp:
      nf[new_hyp] = number of foreign words covered by new_hyp;
      add new_hyp to hypothesisStack[nf[new_hyp]];
      prune hypothesisStack[nf[new_hyp]];
find best hypothesis best_hyp in hypothesisStack[nf];
output best path that leads to best_hyp;
```



SMT, components

Decoder

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.

Wait!

Outline

SMT, the log-linear model

Motivation

Maximum likelihood (ML)

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

Maximum entropy (ME)

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

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

Log-linear model

SMT, the log-linear model

Motivation

Maximum likelihood (ML)

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

Maximum entropy (ME)

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

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

Log-linear model

SMT, the log-linear model

Motivation

Maximum likelihood (ML)

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

Maximum entropy (ME)

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

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

Log-linear model

SMT, the log-linear model

Motivation

Maximum likelihood (ML)

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

Maximum entropy (ME)

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

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

SMT, the log-linear model

Motivation

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

SMT, the log-linear model

Features

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.

SMT, the log-linear model

Features

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

SMT, the log-linear model

Features

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

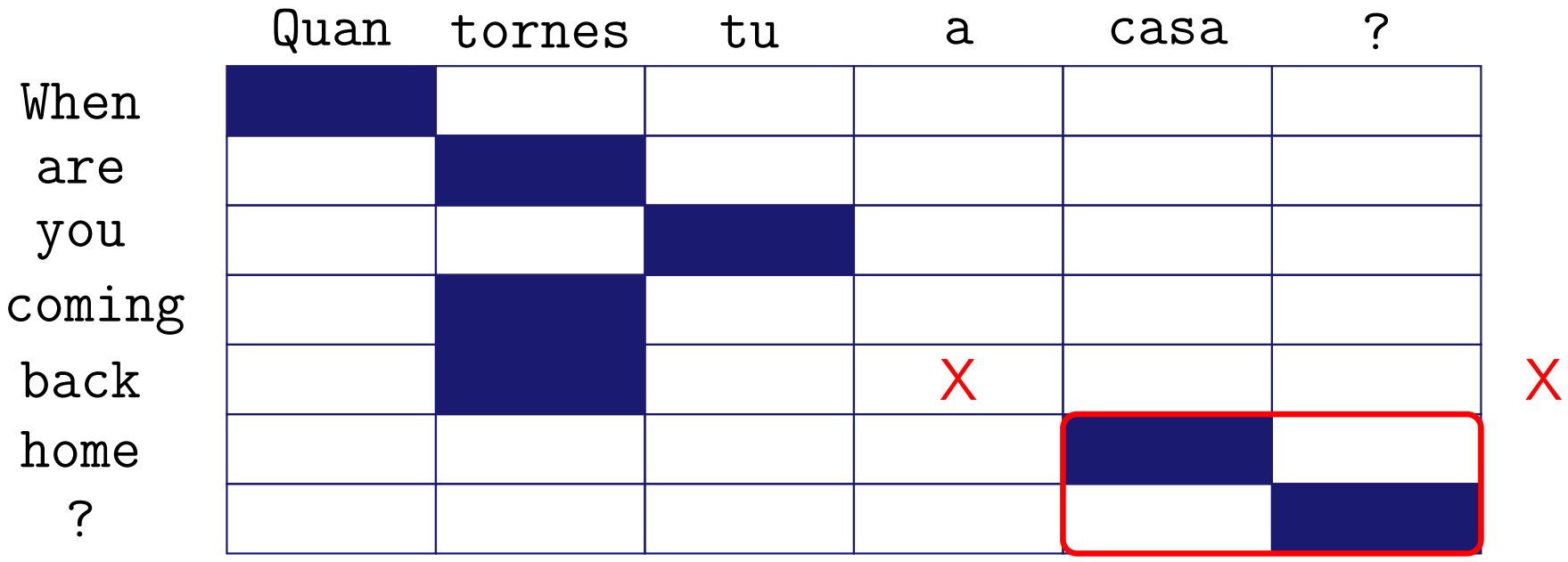
From where and how can one learn reorders?

(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)

SMT, the log-linear model

Digression: lexicalised reordering or distortion

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

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

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

SMT, components

The translation model $P(f|e)$

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
```

SMT, the log-linear model

Features

Standard feature functions

13 features may be used:

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

Wait!

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

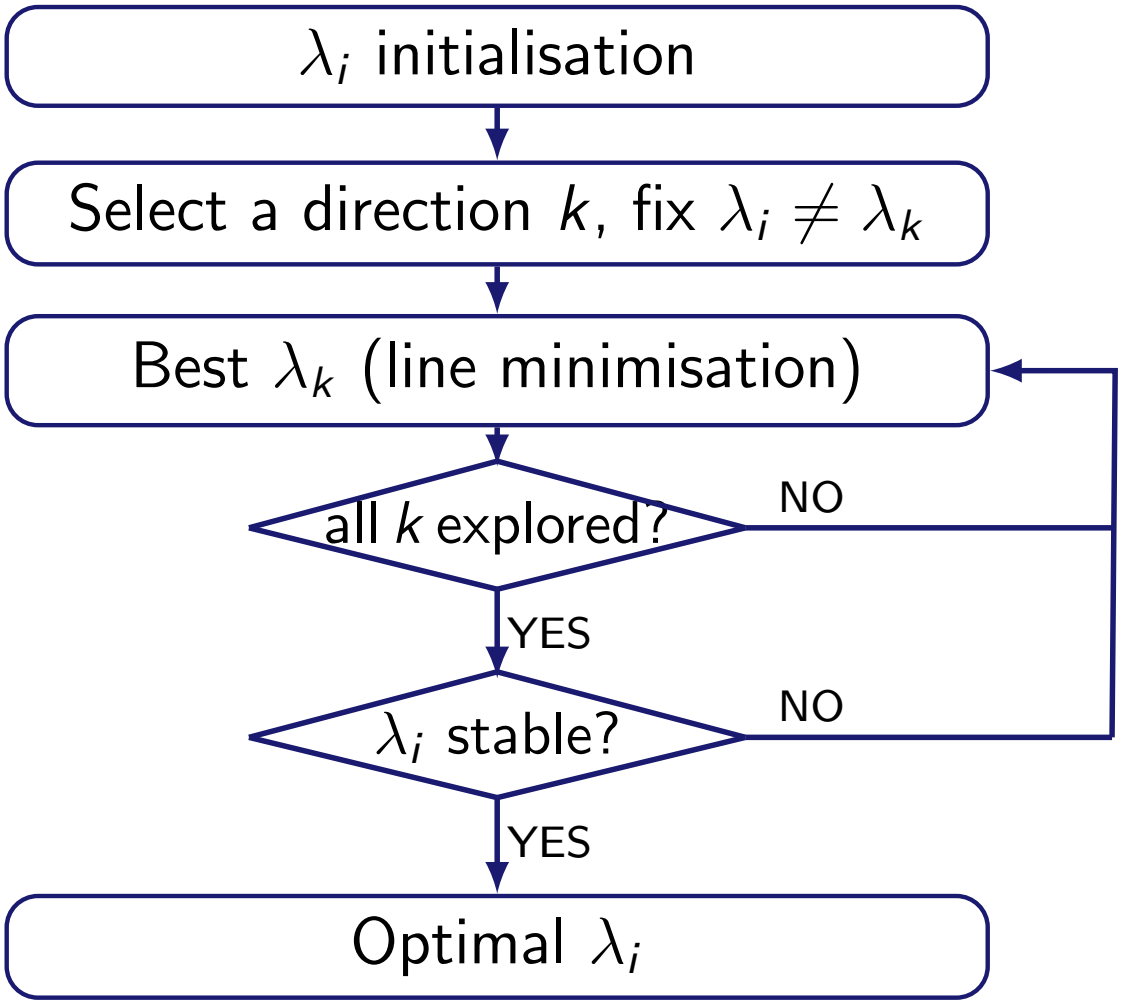
- Approach: Minimise an error function.

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.

(Remember the MT Evaluation section)

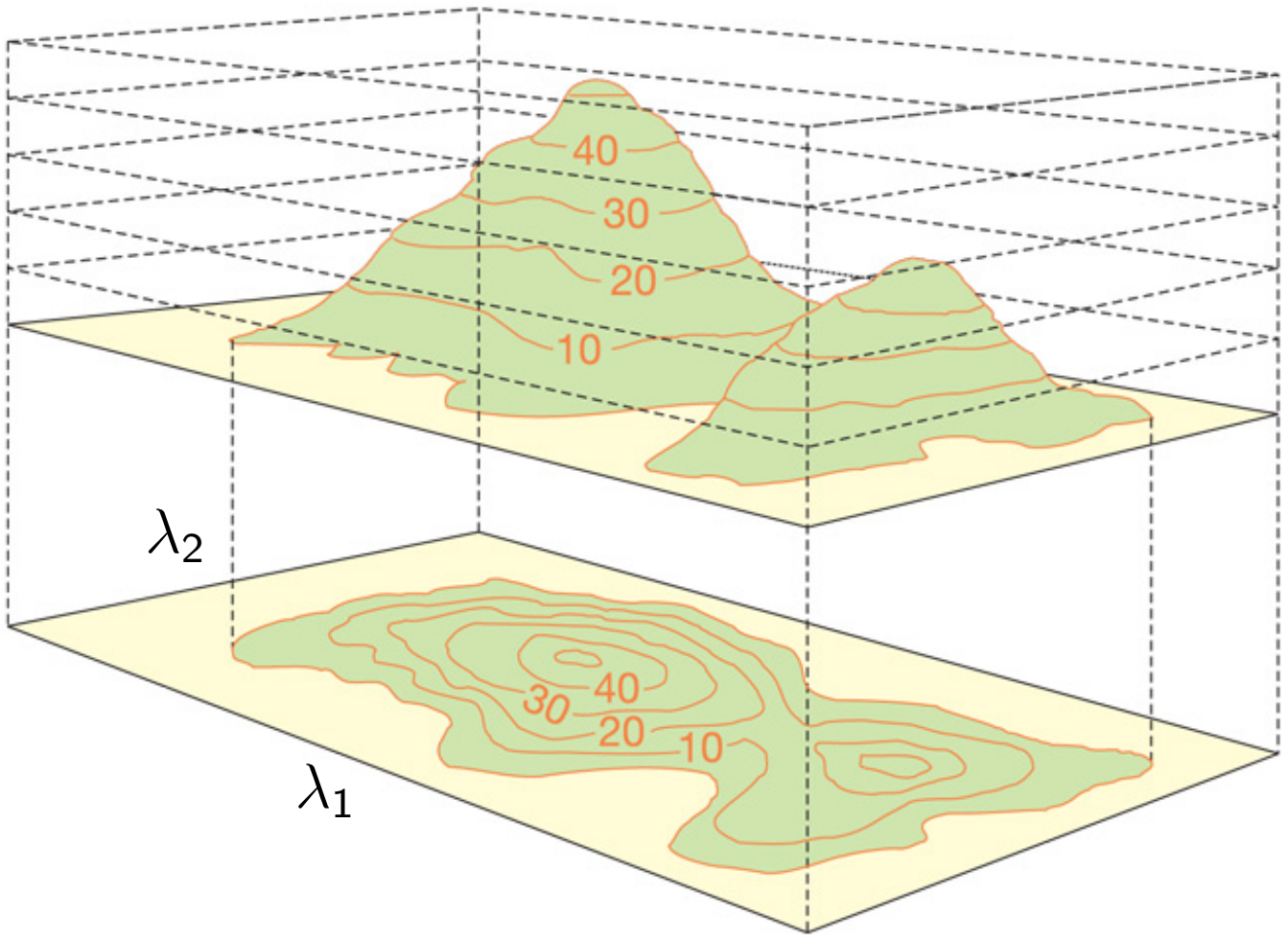
Minimum Error-Rate Training rough algorithm



SMT, the log-linear model

Minimum Error-Rate Training (MERT)

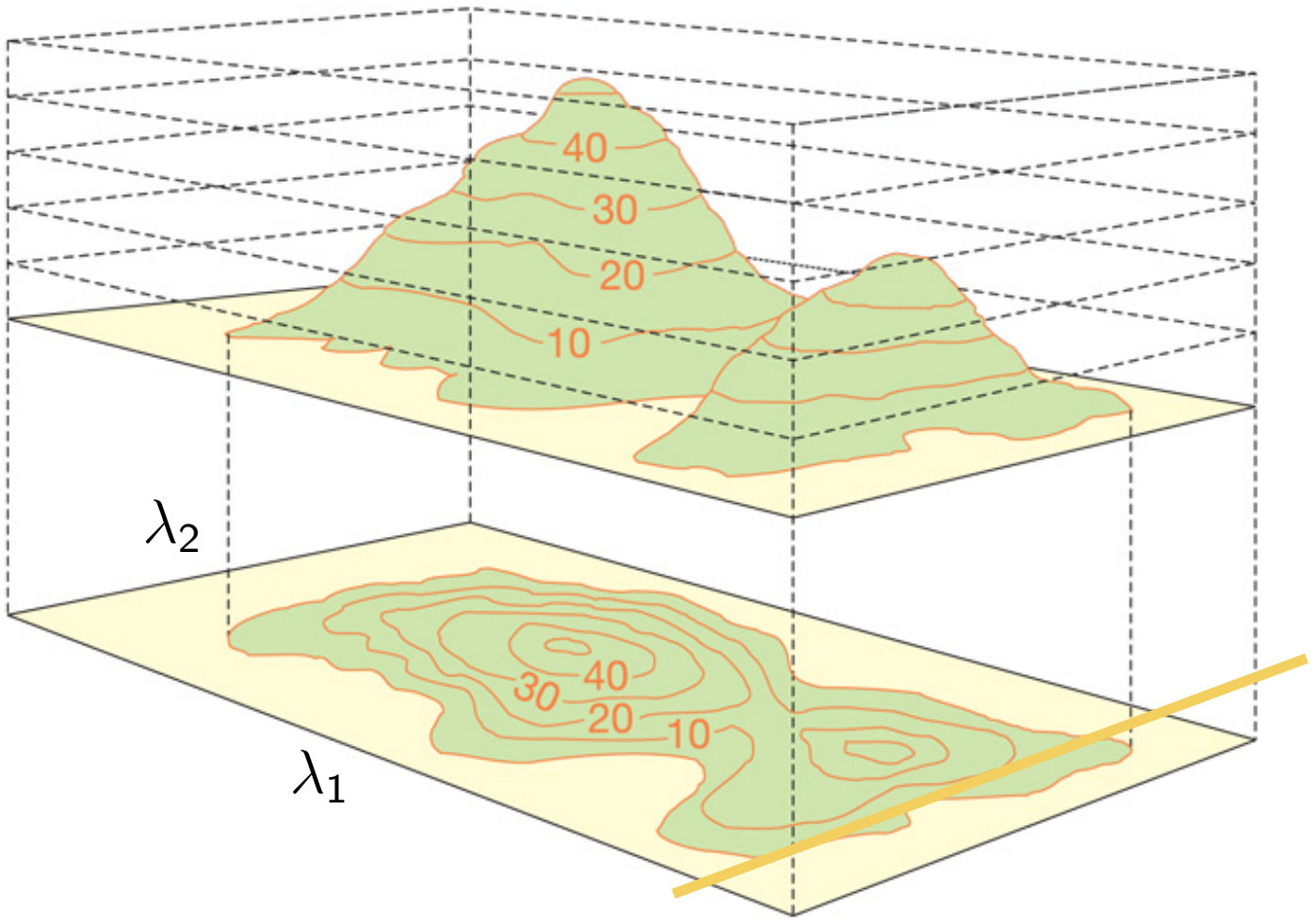
Powell's method (2D: λ_1,λ_2)



SMT, the log-linear model

Minimum Error-Rate Training (MERT)

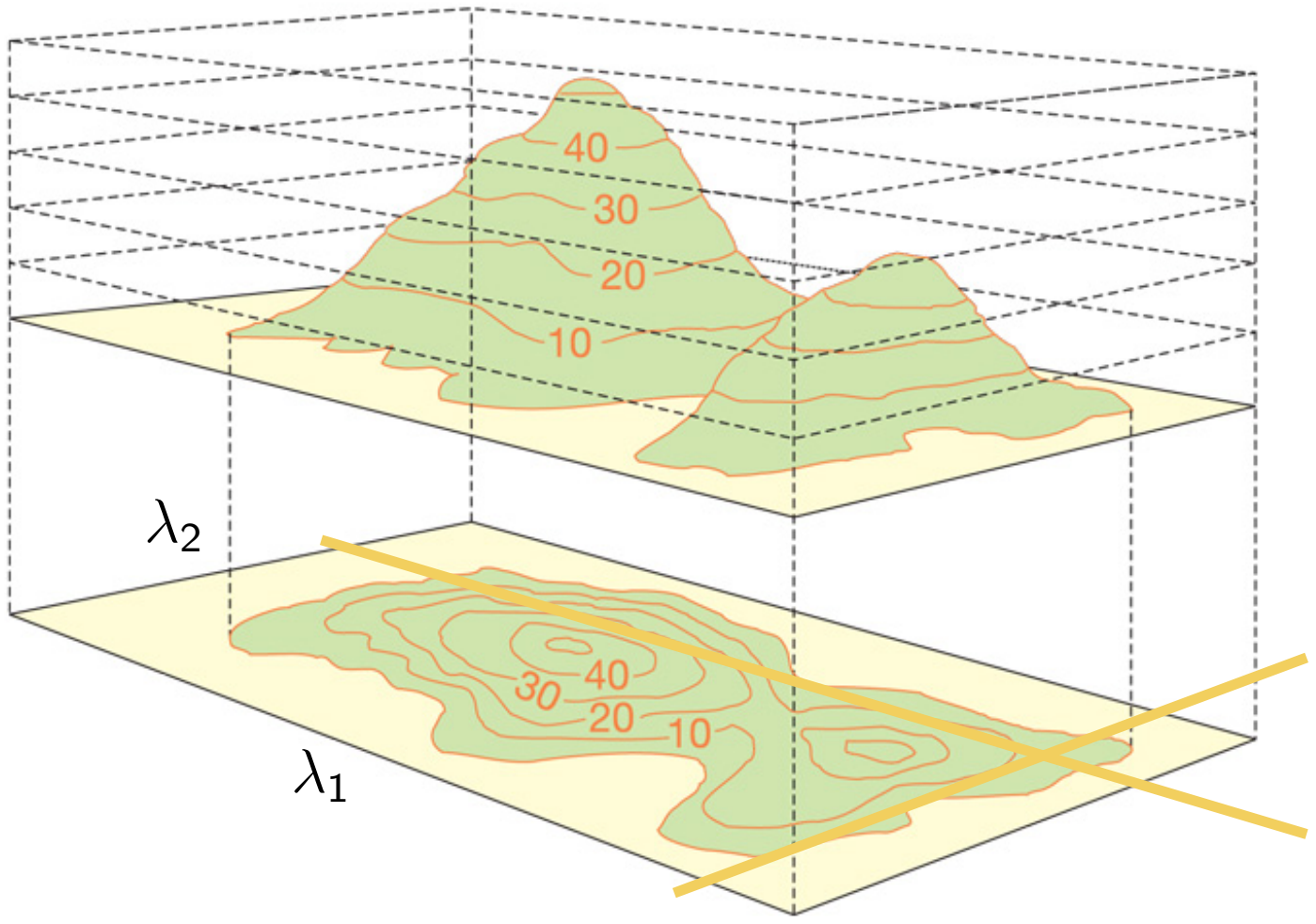
Powell's method (2D: λ_1,λ_2)



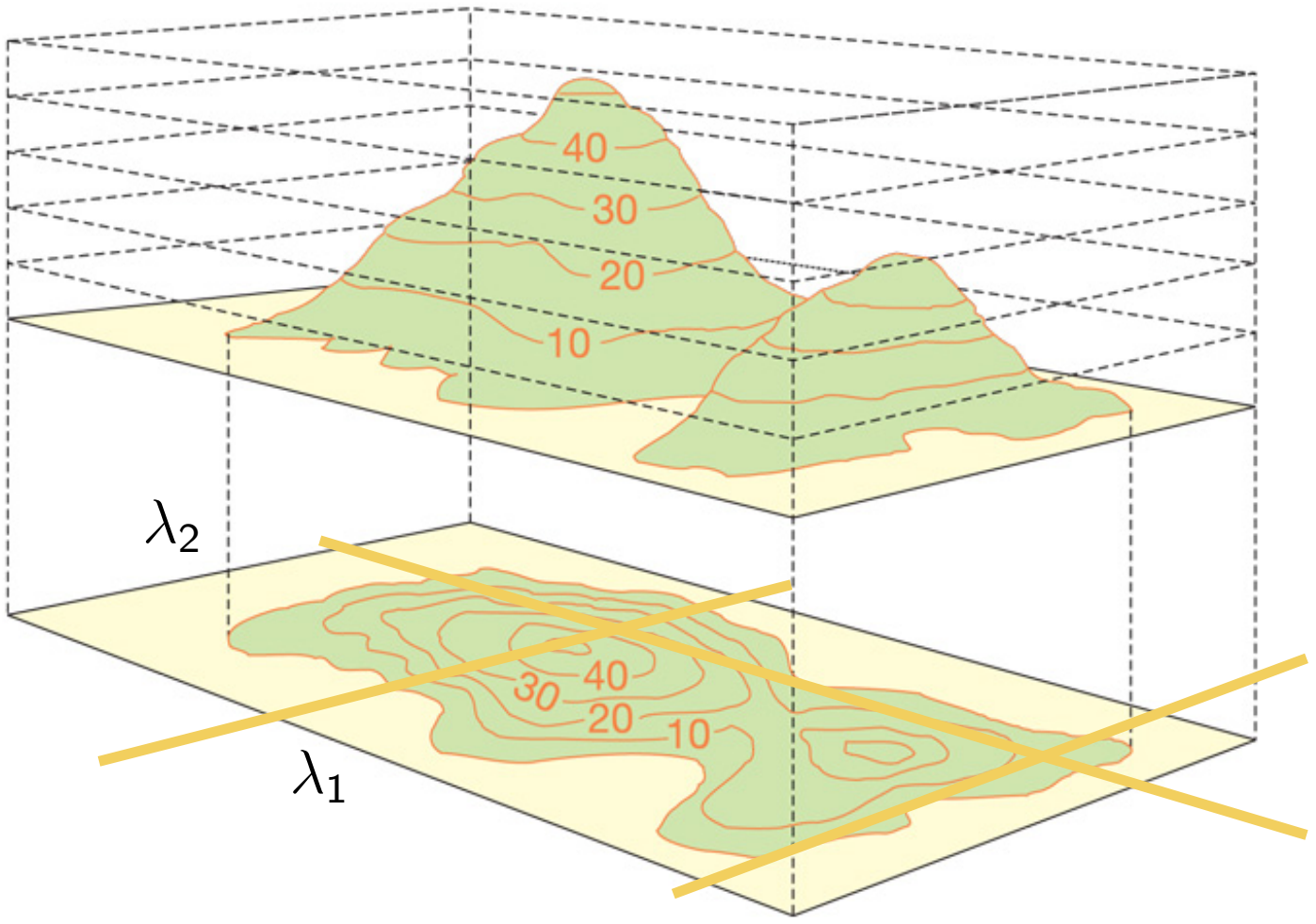
SMT, the log-linear model

Minimum Error-Rate Training (MERT)

Powell's method (2D: λ_1,λ_2)



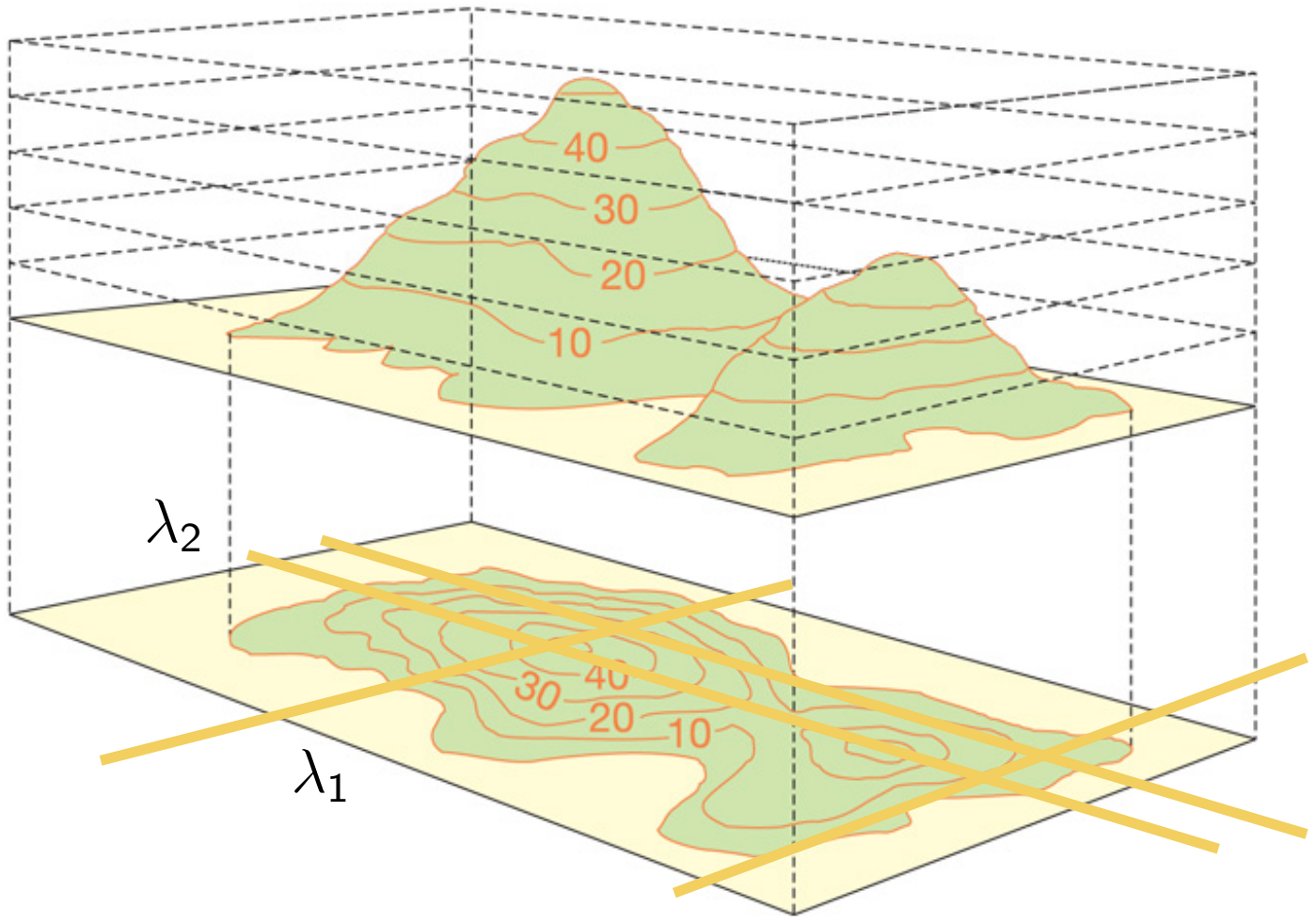
Powell's method (2D: λ_1,λ_2)



SMT, the log-linear model

Minimum Error-Rate Training (MERT)

Powell's method (2D: λ_1,λ_2)



SMT, components

MERT's output

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.

Wait!

Phrase-based SMT systems

Tools & Choices

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`

...

Phrase-based SMT systems

Tools & Choices

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`

...

Phrase-based SMT systems

Tools & Choices

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)

...

Phrase-based SMT systems

Tools & Choices

Decoding with...

Moses

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

Phrasal

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

...

Docent

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

Outline

SMT, beyond standard SMT

Including linguistic information

Considering linguistic information in phrase-based models

- Phrase-based log-linear models do not consider linguistic information other than words. This is information should be included.

Options

- Use syntactic information as pre- or post-process (for reordering or reranking for example).
- Include linguistic information in the model itself.
  - Factored translation models.
  - Syntactic-based translation models.

Factored translation models

Extension to phrase-based models where every word is substituted by a vector of factors.

$$(\texttt{word}) \Longrightarrow (\texttt{word}, \texttt{lemma}, \texttt{PoS}, \texttt{morphology}, \ldots)$$

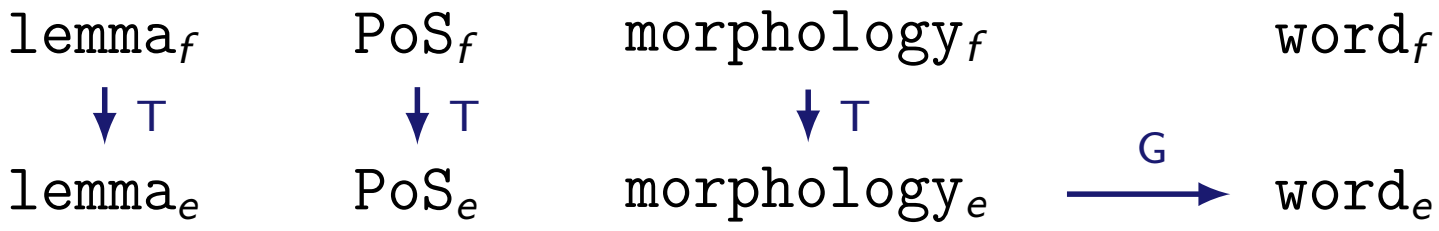
The translation is now a combination of pure translation (T) and generation (G) steps:

Factored translation models

Extension to phrase-based models where every word is substituted by a vector of factors.

$$(\texttt{word}) \Longrightarrow (\texttt{word}, \texttt{lemma}, \texttt{PoS}, \texttt{morphology}, \ldots)$$

The translation is now a combination of pure translation (T) and generation (G) steps:

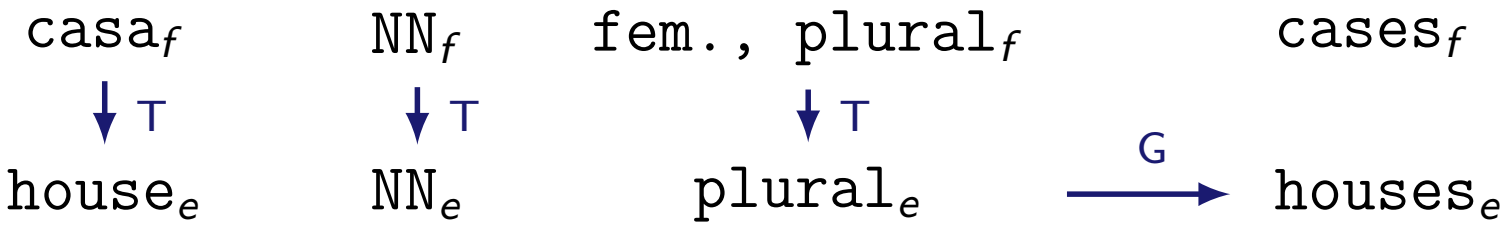


Factored translation models

Extension to phrase-based models where every word is substituted by a vector of factors.

$$(\texttt{word}) \Longrightarrow (\texttt{word}, \texttt{lemma}, \texttt{PoS}, \texttt{morphology}, \ldots)$$

The translation is now a combination of pure translation (T) and generation (G) steps:



What differs in factored translation models
(as compared to standard phrase-based models)

- The parallel corpus must be annotated beforehand
- Extra language models for every factor can also be used
- Translation steps are accomplished in a similar way
- Generation steps imply a training only on the target side of the corpus
- Models corresponding to the different factors and components are combined in a log-linear fashion

What differs in NMT factored translation models
(as compared to SMT factored translation models?)

- SMT: Models corresponding to the different factors and components are combined in a log-linear fashion
- NMT: Models corresponding to the different factors are combined in a concatenating (averaging) the embeddings

Syntactic translation models

Incorporate syntax to the source and/or target languages.

Approaches

- Syntactic phrase-based based on tree trasducers:
  - Tree-to-string. Build mappings from target parse trees to source strings.
  - String-to-tree. Build mappings from target strings to source parse trees.
  - Tree-to-tree. Mappings from parse trees to parse trees.

Syntactic translation models

Incorporate syntax to the source and/or target languages.

Approaches

- Synchronous grammar formalism which learns a grammar that can simultaneously generate both trees.
  - Syntax-based. Respect linguistic units in translation.
  - Hierarchical phrase-based. Respect phrases in translation.

SMT, beyond standard SMT

Syntax-based translation models

Syntactic models ease reordering. An intuitive example:

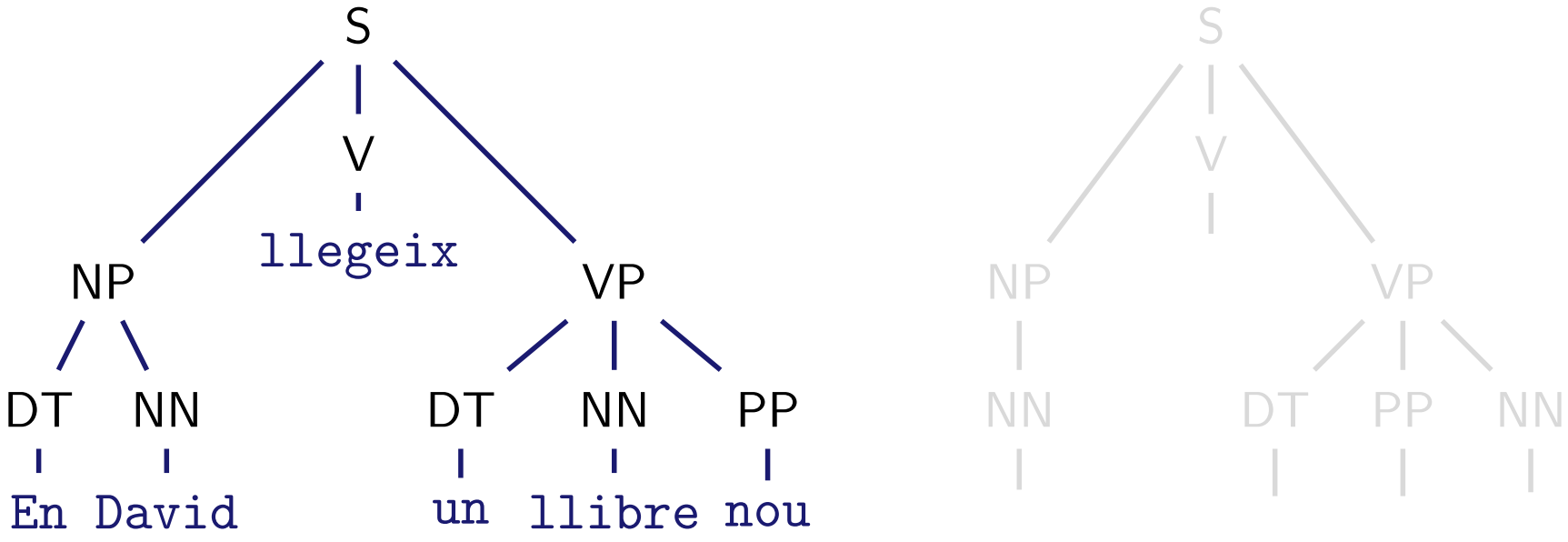
En David llegeix un llibre nou

SMT, beyond standard SMT

Syntax-based translation models

Syntactic models ease reordering. An intuitive example:

En David llegeix un llibre nou

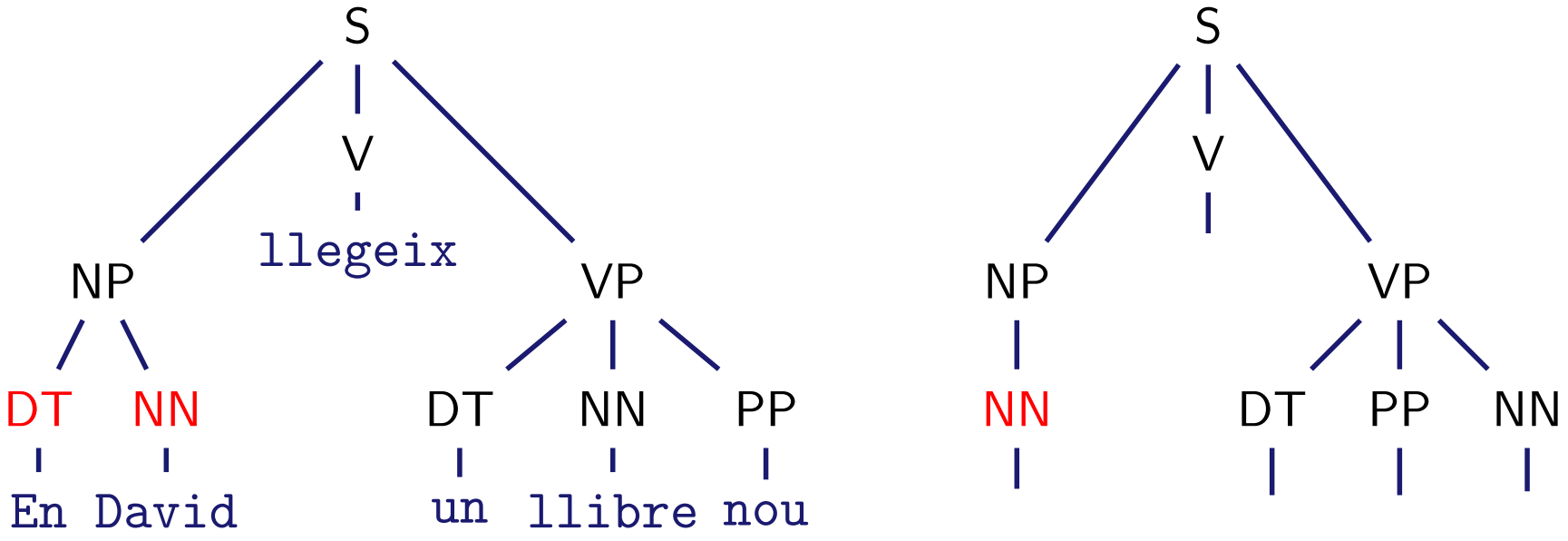


SMT, beyond standard SMT

Syntax-based translation models

Syntactic models ease reordering. An intuitive example:

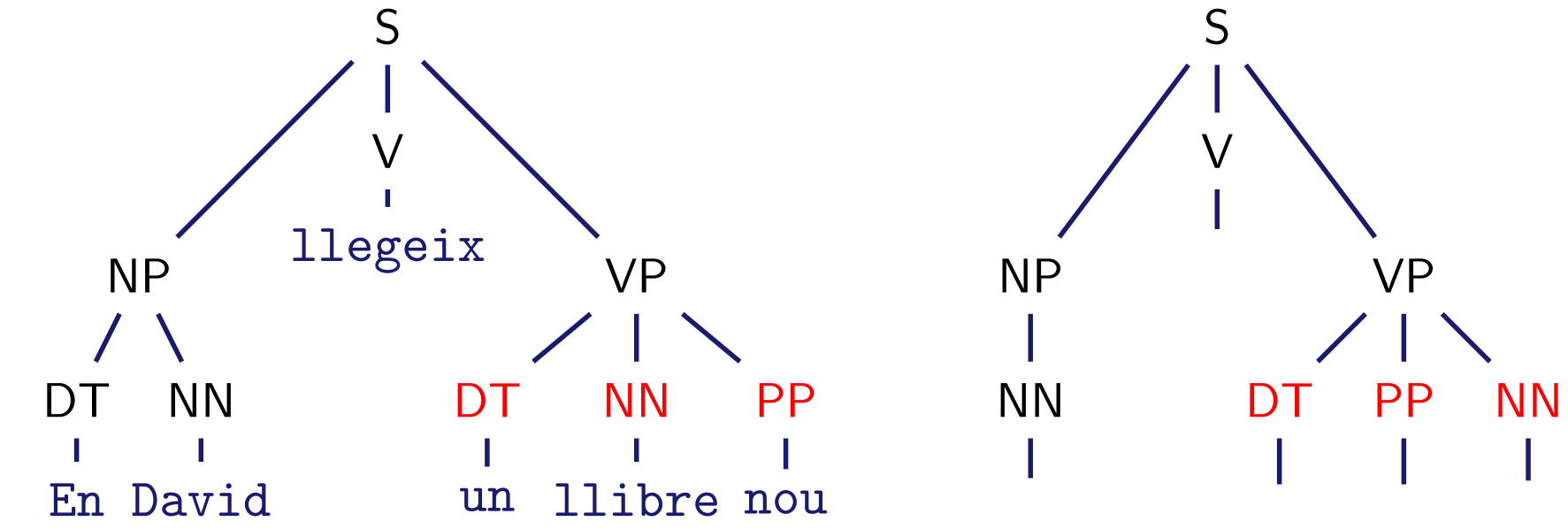
En David llegeix un llibre nou



SMT, beyond standard SMT

Syntax-based translation models

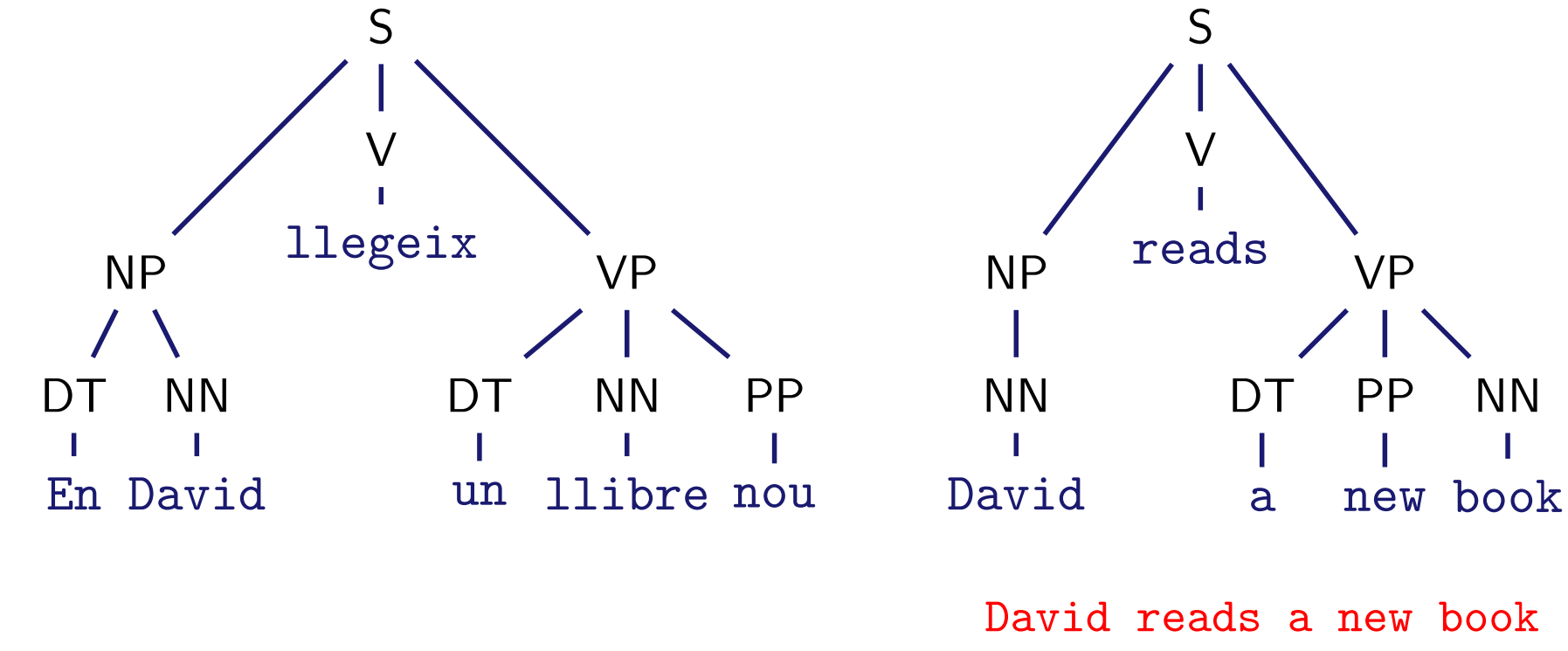
Syntactic models ease reordering. An intuitive example:



SMT, beyond standard SMT

Syntax-based translation models

Syntactic models ease reordering. An intuitive example:



SMT, beyond standard SMT

Including linguistic information

Beyond standard SMT: keep in mind

- Factored models include linguistic information in phrase-based models and are suitable for morphologically rich languages
- Syntactic models consider somehow syntaxis and are adequate for language pairs with a different structure of the sentences
- These extensions have their counterpart in NMT (hybridise!!)

Wait!

Part II

SMT experiments

Outline Part II

6. Translation system
   - Software
   - Steps

Start Preparing!!

Install everything beforehand

Getting Started

The purpose of this lab is to train and evaluate a phrase-based statistical machine translation system (PB-SMT) based on Moses.

First of all you need to install the main software. The process is well documented and there is a very active mailing list where most of the possible installation issues have already been solved by the Moses' developers:
`http://www.statmt.org/moses/?n=Development.GetStarted`
`http://www.statmt.org/moses/?n=Moses.MailingLists`

Tips and Advises

- Do not use Windows
- Read the full installation instructions before starting

Build your own SMT system

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

SMT system

Steps

1. Download and prepare your data
   - Build and clean a corpus
2. Train a Moses PB-SMT engine
   - Analyse the model
3. Evaluate the engine
   - Analyse the outputs
4. Write a report for the lab
   - Intermediate steps and questions regarding extensions to what we have seen in the lectures

Lot of work to do...

Please, come at **16:00** sharp tomorrow!

SMT system

Assignment 1: Lab SMT

Guide & Questionnaire

https://cristinae.github.io/teaching/mt2023/slides/lectures/L1-labSMT.pdf

Assignments at UdS

Policies on Plagiarism and ChatGPT

- Strong policies, we need to report if we discover cases
- Several cases have already been reported (not here :))
- Do the lab yourself, it's not worth cheating!

Assignments at UdS

ChatGPT Writting your Report

CR

What does the heuristic grow-diag-final-and? Which is its purpose?

The heuristic "grow-diag-final-and" (GDF) is a method used in natural language processing for aligning words between two different sequences, such as between a source language sentence and its translation in a target language. The purpose of GDF is to identify the best alignment of words between the two sequences, based on the similarity between the words and the context in which they appear.

The name "grow-diag-final-and" refers to the three main steps involved in the heuristic: "grow", "diag", and "final-and". The "grow" step starts by identifying all possible word pairs between the two sequences that share a similar context. The "diag" step then looks for diagonals in a matrix where the words are compared to each other, identifying a candidate alignment of the two sequences. Finally, the "final-and" step refines the alignment by considering only those word pairs that appear in the diagonal and in the same order in both sequences.

The purpose of GDF is to align the words between two sequences in a way that maximizes their similarity and coherence. This can be useful for a variety of natural language processing tasks, such as machine translation, speech recognition, and text-to-speech synthesis. By aligning the words between the source and target sequences, GDF can help improve the accuracy and fluency of these language processing tasks, making them more effective for real-world applications.

Assignments at UdS

Why I should not use ChatGPT?

- Because you are here to learn, and ChatGPT is not yet integrated into the course
- Because since you are learning you won't be probably able to see mistakes (see previous slide)
- If that's not relevant for you... then because you'll be punished for that
  - and remember that you would be using an NLP tool in a course where there are 3 teachers working on NLP... :p

Part III

Appendix: Classical References

Classical References

History of SMT

- Weaver, 1949 [Wea55]
- Alpac Memorandum [Aut66]
- Hutchins, 1978 [Hut78]
- Slocum, 1985 [Slo85]

The beginnings, word-based SMT

- Brown et al., 1990 [BCP+90]
- Brown et al., 1993 [BPPM93]

Classical References

Phrase-based model

- Och et al., 1999 [OTN99]
- Koehn et al, 2003 [KOM03]

Log-linear model

- Och & Ney, 2002 [ON02]
- Och & Ney, 2004 [ON04]

Factored model

- Koehn & Hoang, 2007 [KH07]

Classical References

Syntax-based models

- Yamada & Knight, 2001 [YK01]
- Chiang, 2005 [Chi05]
- Carreras & Collins, 2009 [CC09]

Discriminative models

- Carpuat & Wu, 2007 [CW07]
- Bangalore et al., 2007 [BHK07]
- Giménez & Màrquez, 2008 [GM08]

Classical References

Language model

- Kneser & Ney, 1995 [KN95]

MERT + alternatives

- Och, 2003 [Och03]
- Neubig & Watanabe, 2016 [NW16]

Domain adaptation

- Bertoldi and Federico, 2009 [Och03]

Classical References

Reordering

- Crego & Mariño, 2006 [Cn06]
- Bach et al., 2009 [BGV09]
- Chen et al., 2009 [CWC09]

Systems combination

- Du et al., 2009 [DMW09]
- Li et al., 2009 [LDZ+09]
- Hildebrand & Vogel, 2009 [HV09]

Classical References

Surveys, theses and tutorials

- Knight, 1999

  http://www.isi.edu/natural-language/mt/wkbk.rtf
- Knight & Koehn, 2003

  http://people.csail.mit.edu/people/koehn/publications/tutorial2003.pdf
- Koehn, 2006

  http://www.iccs.informatics.ed.ac.uk/ pkoehn/publications/tutorial2006.pdf
- Way & Hassan, 2009

  http://www.medar.info/conference_all/2009/Tutorial_3.pdf
- Lopez, 2008 [Lop08]
- Giménez, 2009 [Gim08]

Classical References I

Automatic Language Processing Advisory Committee (ALPAC).
Language and Machines. Computers in Translation and Linguistics.
Technical Report Publication 1416, Division of Behavioural Sciences, National Academy of Sciences, National Research Council, Washington, D.C., 1966.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin.
A statistical approach to machine translation.
Computational Linguistics, 16(2):79–85, 1990.

Nguyen Bach, Qin Gao, and Stephan Vogel.
Source-side dependency tree reordering models with subtree movements and constraints.
In *Proceedings of the Twelfth Machine Translation Summit (MTSummit-XII)*, Ottawa, Canada, August 2009. International Association for Machine Translation.

Srinivas Bangalore, Patrick Haffner, and Stephan Kanthak.
Statistical Machine Translation through Global Lexical Selection and Sentence Reconstruction.
In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 152–159, 2007.

Classical References II

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

Xavier Carreras and Michael Collins.
Non-projective parsing for statistical machine translation.
In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 200–209, Singapore, August 2009.

David Chiang.
A hierarchical phrase-based model for statistical machine translation.
In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

Josep M^a Crego and José B. Mari no.
Improving smt by coupling reordering and decoding.
Machine Translation, 20(3):199–215, March 2006.

Classical References III

Marine Carpuat and Dekai Wu.
Improving Statistical Machine Translation Using Word Sense Disambiguation.
In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–72, 2007.

Han-Bin Chen, Jian-Cheng Wu, and Jason S. Chang.
Learning bilingual linguistic reordering model for statistical machine translation.
In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 254–262, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

Jinhua Du, Yanjun Ma, and Andy Way.
Source-side context-informed hypothesis alignment for combining outputs from Machine Translation systems.
In *Proceedings of the Machine Translation Summit XII*, pages 230–237, Ottawa, ON, Canada., 2009.

Jesś Giménez.
Empirical Machine Translation and its Evaluation.
PhD thesis, Universitat Politècnica de Catalunya, July 2008.

Classical References IV


Jesús Giménez and Lluís Màrquez.
Discriminative Phrase Selection for SMT, pages 205–236.
NIPS Workshop Series. MIT Press, 2008.


W. J. Hutchins.
Machine translation and machine-aided translation.
Journal of Documentation, 34(2):119–159, 1978.


Almut Silja Hildebrand and Stephan Vogel.
CMU system combination for WMT'09.
In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 47–50, Athens, Greece, March 2009. Association for Computational Linguistics.


Philipp Koehn and Hieu Hoang.
Factored Translation Models.
In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 868–876, 2007.


R. Kneser and H. Ney.
Improved backing-off for m-gram language modeling.
icassp, 1:181–184, 1995.

Classical References V

Philipp Koehn, Franz Josef Och, and Daniel Marcu.
Statistical phrase-based translation.
In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Edomonton, Canada, May 27-June 1 2003.

Mu Li, Nan Duan, Dongdong Zhang, Chi-Ho Li, and Ming Zhou.
Collaborative decoding: Partial hypothesis re-ranking using translation consensus between decoders.
In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 585–592, Suntec, Singapore, August 2009. Association for Computational Linguistics.

Adam Lopez.
Statistical machine translation.
ACM Comput. Surv., 40(3), 2008.

Graham Neubig and Taro Watanabe.
Optimization for statistical machine translation: A survey.
Computational Linguistics, 42(1):1–54, March 2016.

Classical References VI


Franz Josef Och.
Minimum error rate training in statistical machine translation.
In *Proc. of the Association for Computational Linguistics*, Sapporo, Japan, July 6-7 2003.


Franz Josef Och and Hermann Ney.
Discriminative Training and Maximum Entropy Models for Statistical Machine Translation.
In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 295–302, 2002.


Franz Josef Och and Hermann Ney.
The alignment template approach to statistical machine translation.
Computational Linguistics, 30(4):417–449, 2004.


Franz Josef Och, Christoph Tillmann, and Hermann Ney.
Improved alignment models for statistical machine translation.
In *Proc. of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, University of Maryland, College Park, MD, June 1999.

Classical References VII

Jonathan Slocum.
A survey of machine translation: its history, current status, and future prospects.
Comput. Linguist., 11(1):1–17, 1985.

Warren Weaver.
Translation.
In William N. Locke and A. Donald Boothe, editors, *Machine Translation of Languages*, pages 15–23. MIT Press, Cambridge, MA, 1949/1955.
Reprinted from a memorandum written by Weaver in 1949.

Kenji Yamada and Kevin Knight.
A syntax-based statistical translation model.
In *Proceedings of the 39rd Annual Meeting of the Association for Computational Linguistics (ACL'01)*, Toulouse, France, July 2001.