Practical introduction to Statistical Machine Translation

Cristina España i Bonet

GF Summer School

23rd August 2011

Overview

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

Part I: SMT background

$\sim$ 90min

Overview

6. Translation system
7. MT Evaluation basics
8. Evaluation system

Part II: SMT experiments

from 30min to...

Part III: MT evaluation

$\sim$ 1h

Part I

SMT background

Outline

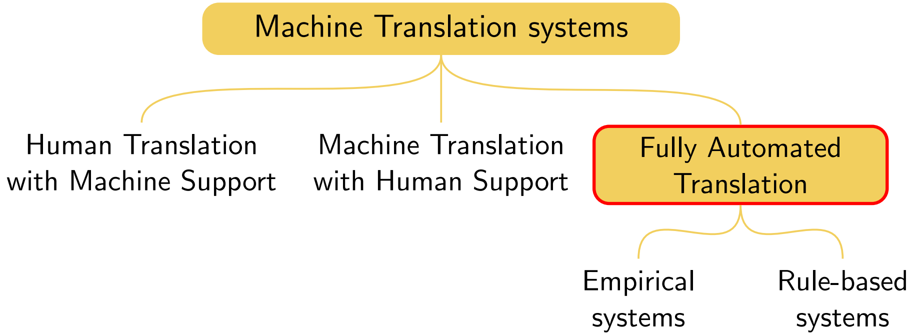
1. Introduction
2. Basics
3. Components
4. The log-linear model
5. Beyond standard SMT

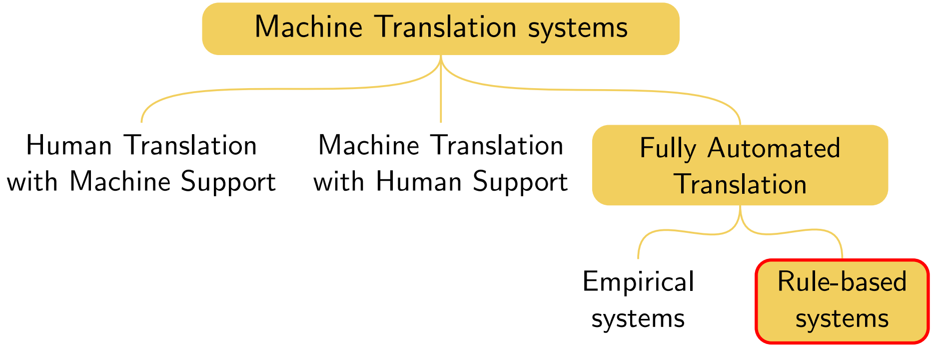
Introduction

Machine Translation Taxonomy

Machine Translation systems

- Human Translation with Machine Support
- Machine Translation with Human Support
- Fully Automated Translation

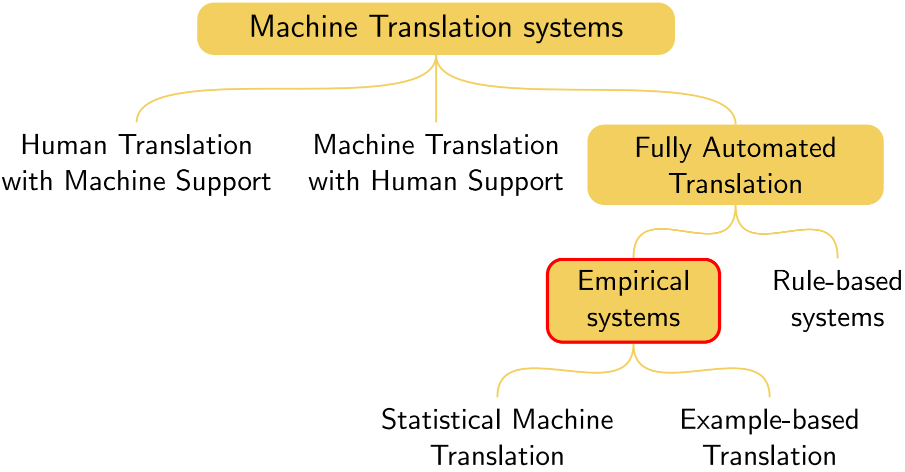

Machine Translation systems
Human Translation with Machine Support
Machine Translation with Human Support
Fully Automated Translation
Empirical systems
Rule-based systems



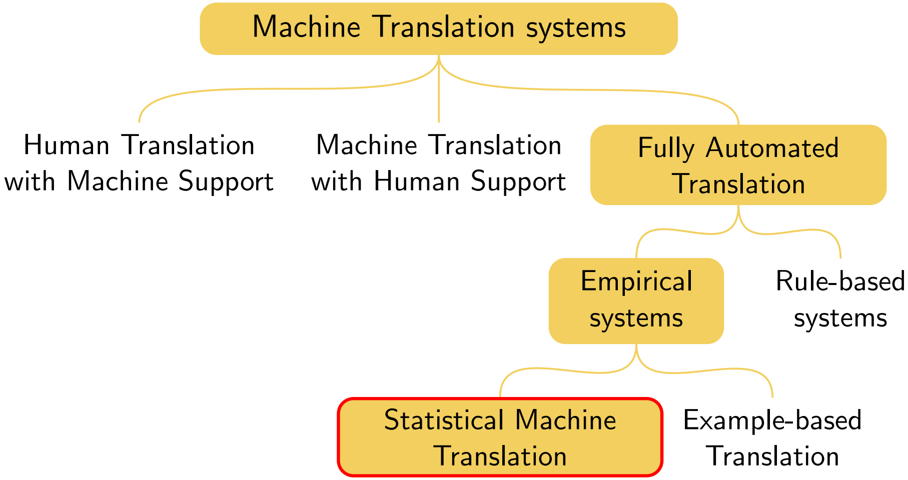
Machine Translation systems
Human Translation with Machine Support
Machine Translation with Human Support
Fully Automated Translation
Empirical systems
Rule-based systems

Introduction

Machine Translation Taxonomy

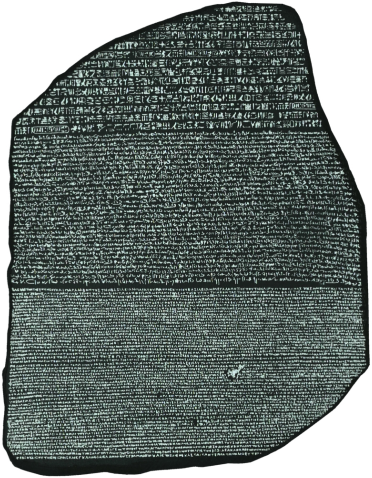


Introduction
Machine Translation Taxonomy

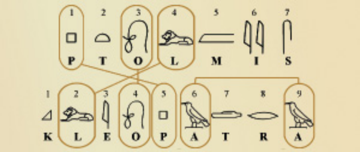


Introduction

Empirical Machine Translation



Empirical MT relies on aligned corpora.



Empirical MT relies on large parallel aligned corpora.

Com a tècnica principal, MOLTO utilitza gramàtiques semàntiques de domini específic i interlingues basades en ontologies. Aquests components s'implementen en GF (Grammatical Framework), un formalisme de gramàtiques on es relacionen diversos idiomes a través d'una sintaxi abstracta comú. El GF s'ha aplicat en diversos dominis de mida petita i mitjana, típicament per tractar fins a un total de deu idiomes, però MOLTO ampliarà això en termes de productivitat i aplicabilitat.

As its main technique, MOLTO uses domain-specific semantic grammars and ontology-based interlinguas. These components are implemented in GF (Grammatical Framework), which is a grammar formalism where multiple languages are related by a common abstract syntax. GF has been applied in several small-to-medium size domains, typically targeting up to ten languages but MOLTO will scale this up in terms of productivity and applicability.

Empirical MT relies on large parallel aligned corpora.

Com a tècnica principal, MOLTO utilitza gramàtiques semàntiques de domini específic i interlingues basades en ontologies. Aquests components s'implementen en GF (Grammatical Framework), un formalisme de gramàtiques on es relacionen diversos idiomes a través d'una sintaxi abstracta comú. El GF s'ha aplicat en diversos dominis de mida petita i mitjana, típicament per tractar fins a un total de deu idiomes, però MOLTO ampliarà això en termes de productivitat i aplicabilitat.

As its main technique, MOLTO uses domain-specific semantic grammars and ontology-based interlinguas. These components are implemented in GF (Grammatical Framework), which is a grammar formalism where multiple languages are related by a common abstract syntax. GF has been applied in several small-to-medium size domains, typically targeting up to ten languages but MOLTO will scale this up in terms of productivity and applicability.

Empirical MT relies on **large parallel** aligned corpora.

L'objectiu de MOLTO és desenvolupar un conjunt d'eines per a traduir textos entre diversos idiomes en temps real i amb alta qualitat. Les llengües són mòduls separats en l'eina i per tant es poden canviar; els prototips que es construiran cobriran la major part dels 23 idiomes oficials de la UE.

Com a tècnica principal, MOLTO utilitza gramàtiques semàntiques de domini específic i interlingues basades en ontologies. Aquests components s'implementen en GF (Grammatical Framework), un formalisme de gramàtiques on es relacionen diversos idiomes a través d'una sintaxi abstracta comú. El GF s'ha aplicat en diversos dominis de mida petita i mitjana, típicament per tractar fins a un total de deu idiomes, però MOLTO ampliarà això en termes de productivitat i aplicabilitat.

Part de l'ampliació es dedicarà a augmentar la mida dels dominis i el nombre d'idiomes. Una part important és fer la tecnologia accessible per als experts del domini sense experiència amb GFs i reduir al mínim l'esforç necessari per a la construcció d'un traductor. Idealment, això es pot fer només estenent un lexicó i escrivint un conjunt de frases d'exemple.

MOLTO's goal is to develop a set of tools for translating texts between multiple languages in real time with high quality. languages are separate modules in the tool and can be varied; prototypes covering a majority of the EU's 23 official languages will be built.

As its main technique, MOLTO uses domain-specific semantic grammars and ontology-based interlinguas. These components are implemented in GF (Grammatical Framework), which is a grammar formalism where multiple languages are related by a common abstract syntax. GF has been applied in several small-to-medium size domains, typically targeting up to ten languages but MOLTO will scale this up in terms of productivity and applicability.

A part of the scale-up is to increase the size of domains and the number of languages. A more substantial part is to make the technology accessible for domain experts without GF expertise and minimize the effort needed for building a translator. Ideally, this can be done by just extending a lexicon and writing a set of example sentences.

Aligned parallel corpora numbers

Corpora

Corpus	# segments (app.)	# words (app.)
JRC-Acquis	$1.0 \cdot 10^6$	$30 \cdot 10^6$
Europarl	$1.5 \cdot 10^6$	$45 \cdot 10^6$
United Nations	$3.8 \cdot 10^6$	$100 \cdot 10^6$

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	$1.5 \cdot 10^6$	$45 \cdot 10^6$
United Nations	$3.8 \cdot 10^6$	$100 \cdot 10^6$

Books

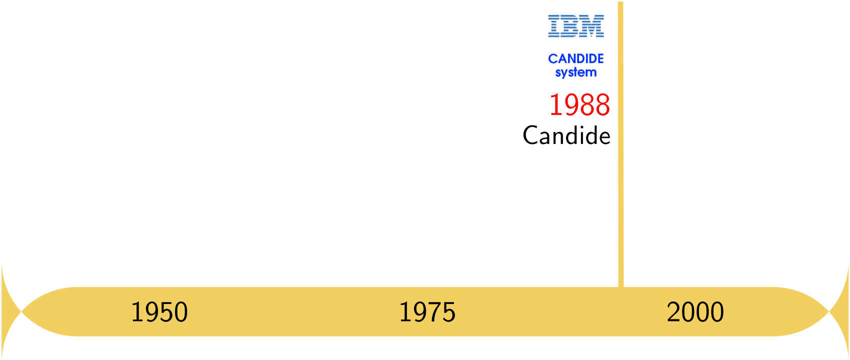
Title	# words (approx.)
The Bible	$0.8 \cdot 10^6$
The Dark Tower series	$1.2 \cdot 10^6$
Encyclopaedia Britannica	$44 \cdot 10^6$

Outline

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

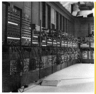
SMT, basics

The beginnings, summarised timeline

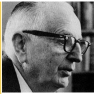


SMT, basics

The beginnings, summarised timeline



1946
ENIAC



1949
Weaver memo

1966
Alpac report

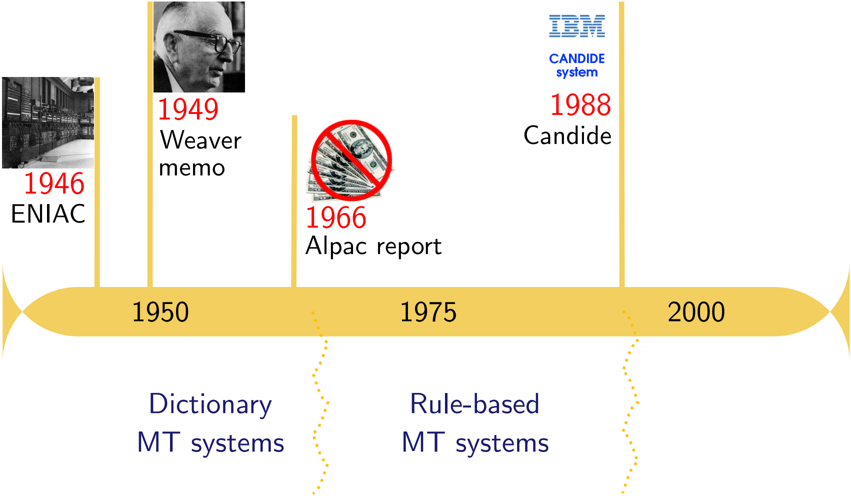
CANDIDE system

1988
Candide

1950 1975 2000

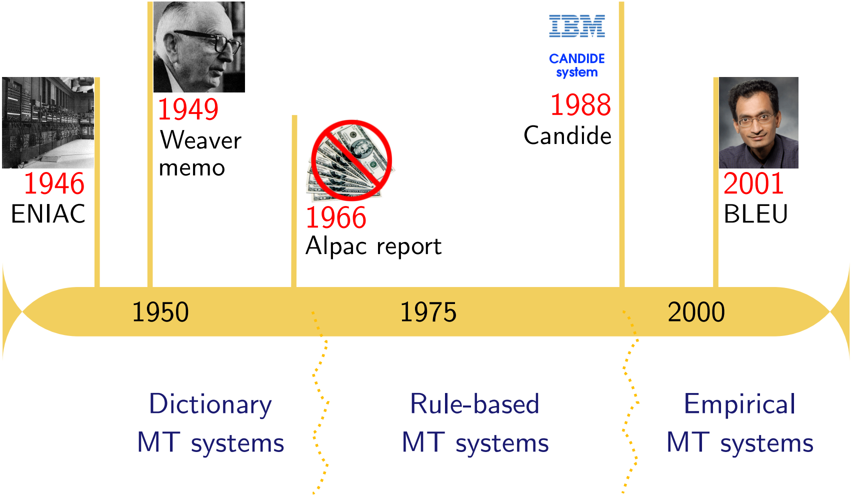
SMT, basics

The beginnings, summarised timeline



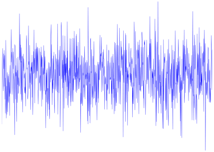
SMT, basics

The beginnings, summarised timeline



The Noisy Channel as a statistical approach to translation:

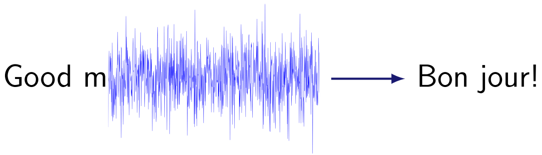
Good morning! →



SMT, basics

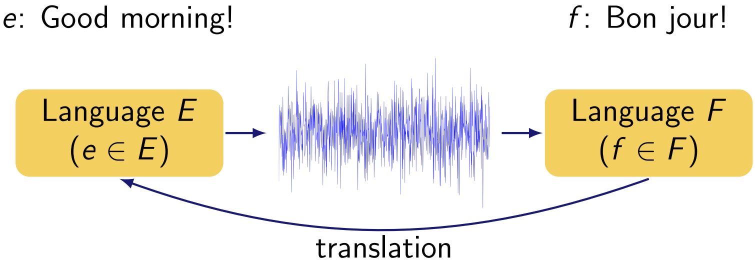
The Noisy Channel approach

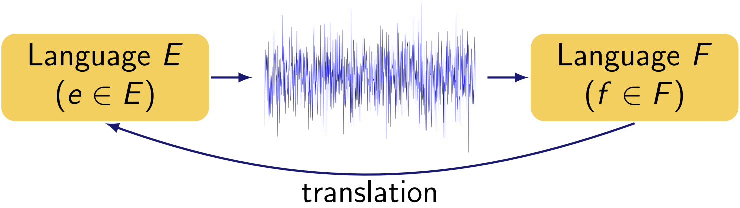
The Noisy Channel as a statistical approach to translation:



SMT, basics
The Noisy Channel approach

The Noisy Channel as a statistical approach to translation:

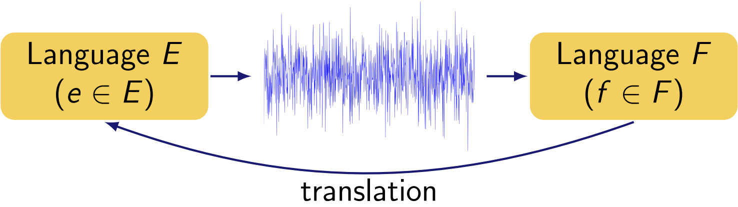




Mathematically:

$$P(e|f)$$

SMT, basics
The Noisy Channel approach



Mathematically:

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

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

SMT, basics

Components

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

Language Model

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

Translation Model

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

argmax

- Search done by the *decoder*

SMT, basics

Components

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

Language Model

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

Translation Model

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

argmax

- Search done by the *decoder*

SMT, basics

Components

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

Language Model

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

Translation Model

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

argmax

- Search done by the *decoder*

Outline

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

SMT, components

The language model $P(e)$

Language model

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

Estimation of how probable a sentence is.

Naïve estimation on a corpus with N sentences:

Frequentist probability of a sentence e:

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

Problem:

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

Language model

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

Estimation of how probable a sentence is.

Naïve estimation on a corpus with N sentences:

Frequentist probability of a sentence e:

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

Problem:

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

Language model

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

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.

Example, a 4-gram model

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

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

where, for each factor,

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

SMT, components

The language model $P(e)$

Example, a 4-gram model

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

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

where, for each factor,

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

Example, a 4-gram model

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

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

where, for each factor,

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

Example, a 4-gram model

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

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

where, for each factor,

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

Example, a 4-gram model

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

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

where, for each factor,

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

SMT, components

The language model $P(e)$

Example, a 4-gram model

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

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

where, for each factor,

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

Example, a 4-gram model

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

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

where, for each factor,

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

SMT, components

The language model $P(e)$

Example, a 4-gram model

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

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

where, for each factor,

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

Example, a 4-gram model

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

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

where, for each factor,

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

Example, a 4-gram model

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

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

where, for each factor,

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

SMT, components

The language model $P(e)$

Example, a 4-gram model

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

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

where, for each factor,

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

SMT, components

The language model $P(e)$

Example, a 4-gram model

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

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

where, for each factor,

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

SMT, components

The language model $P(e)$

Main problems and criticisims:

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

Solution

Smoothing techniques:

- Linear interpolation.

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

SMT, components

The language model $P(e)$

Main problems and criticisims:

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

Solution

Smoothing techniques:

- Linear interpolation.
- Back-off models.

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

SMT, components

The language model $P(e)$

Main problems and criticisims:

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

Solution

Smoothing techniques:

- Linear interpolation.

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

Main problems and criticisims:

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

Solution

Smoothing techniques:

- Linear interpolation.

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

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.

SMT, components

The translation model $P(f|e)$

Translation model

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

Estimation of the lexical correspondence between languages.

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

SMT, components

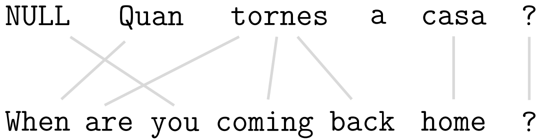
The translation model $P(f|e)$

Translation model

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

Estimation of the lexical correspondence between languages.

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



SMT, components

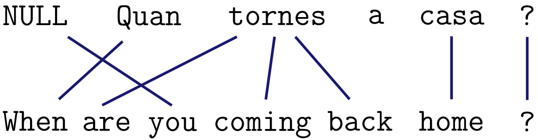
The translation model $P(f|e)$

Translation model

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

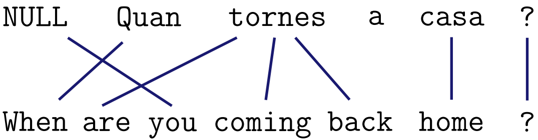
Estimation of the lexical correspondence between languages.

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



SMT, components

The translation model $P(f|e)$



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

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

Word-based models: the IBM models

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

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

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

Word-based models: the IBM models

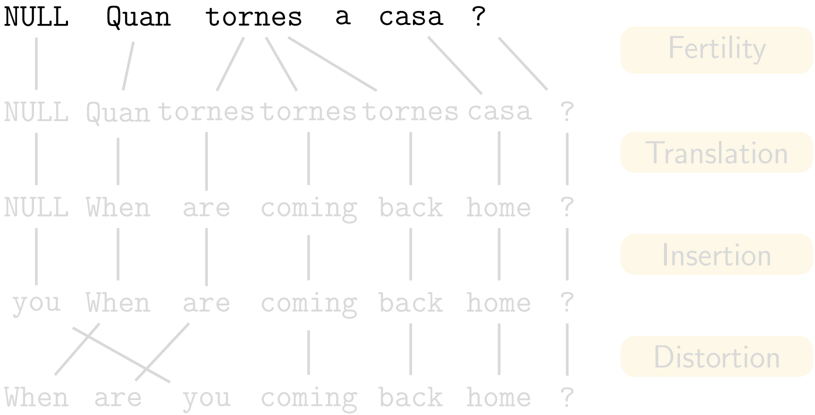
They characterise $P(f|e)$ with 4 parameters: t, n, d and p_1.

- 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 $P(f|e)$

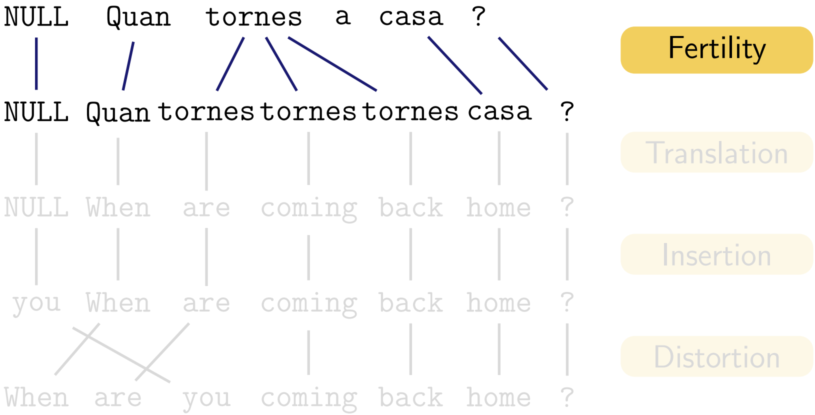
Back to the example:



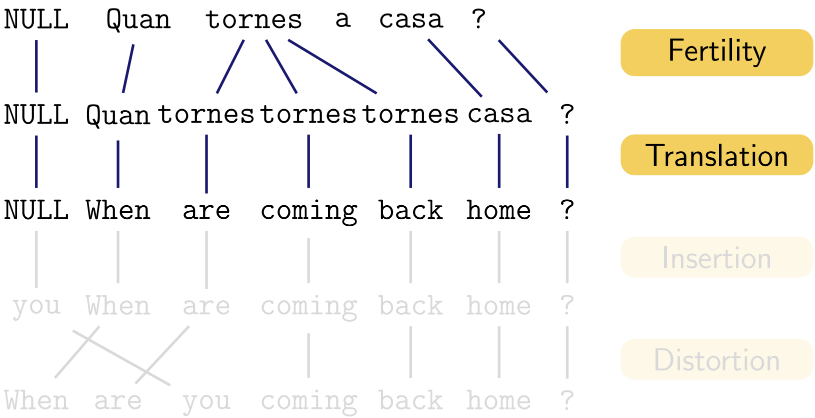
SMT, components

The translation model $P(f|e)$

Back to the example:



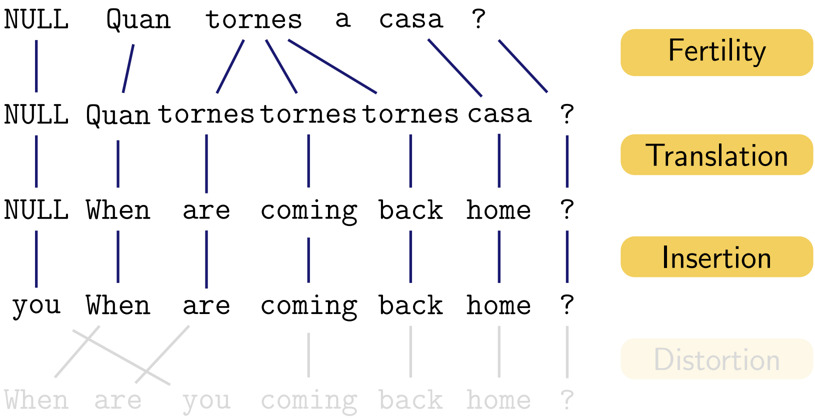
Back to the example:



SMT, components

The translation model $P(f|e)$

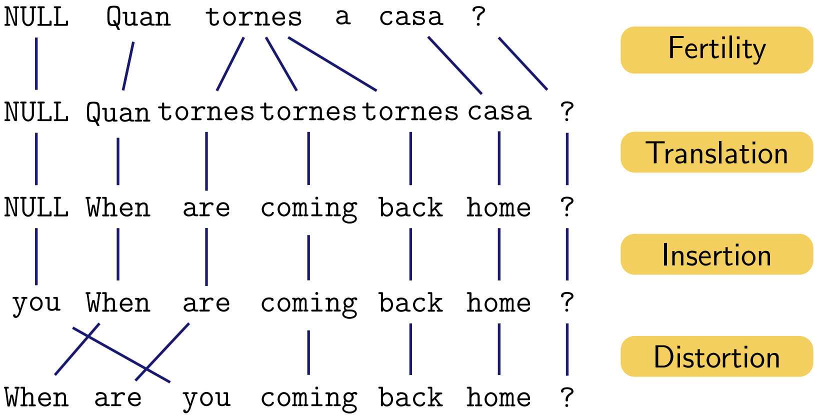
Back to the example:



SMT, components

The translation model $P(f|e)$

Back to the example:



Word-based models: the IBM models

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

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

But...

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

Solutions

- 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 be t, n, d and p_1 estimated?

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

But...

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

Solutions

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

Word-based models: the IBM models

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

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

But...

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

Solutions

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

SMT, components

The translation model $P(f|e)$

Expectation-Maximisation algorithm

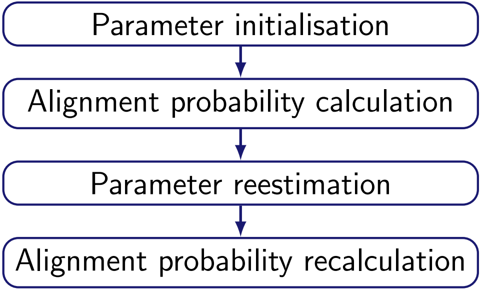
Parameter initialisation

Alignment probability calculation

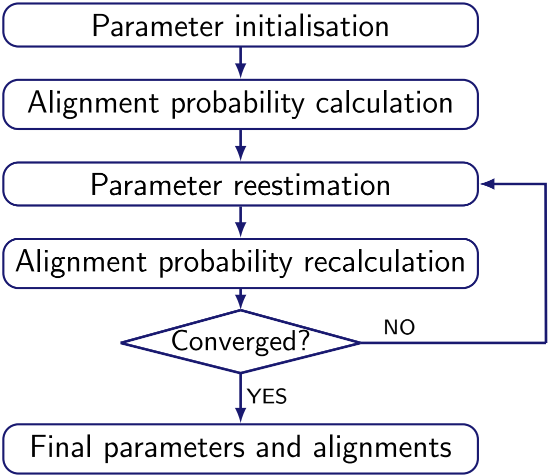
SMT, components

The translation model $P(f|e)$

Expectation-Maximisation algorithm



Expectation-Maximisation algorithm

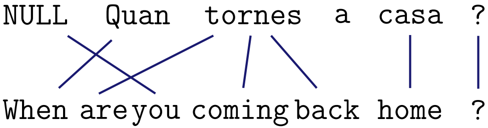


Alignment's asymmetry

The definitions in IBM models make the alignments asymmetric

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

Catalan to English



English to Catalan

SMT, components

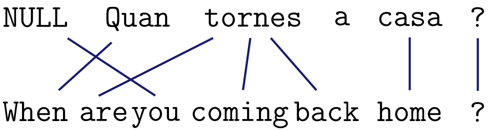
The translation model $P(f|e)$

Alignment's asymmetry

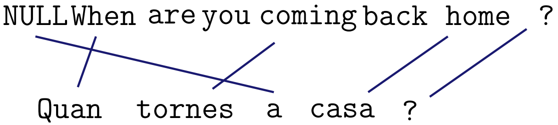
The definitions in IBM models make the alignments asymmetric

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

Catalan to English



English to Catalan



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

Graphically:

Catalan to English

SMT, components

The translation model $P(f|e)$

Graphically:

English to Catalan

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

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: ϕ

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the book`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the book new.`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

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

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

e: ϕ

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

e: `David`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

e: `David reads`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

e: `David reads the`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

e: `David reads the book`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

e: `David reads the book of`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

e: `David reads the book of new.`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

e: `David reads the book of new.` ✗

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

e: `David reads the book of new.` ✗

e: ϕ

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

e: `David reads the book of new.` ✗

e: `David`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

e: `David reads the book of new.` ✗

e: `David reads`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

e: `David reads the book of new.` ✗

e: `David reads the`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

e: `David reads the book of new.` ✗

e: `David reads the book`

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

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

From Word-based to Phrase-based models

f: `En David llegeix el llibre nou.`

e: `David reads the new book.` ✓

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

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

From Word-based to Phrase-based models

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

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

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

e: `David reads the book of new.` ✗

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

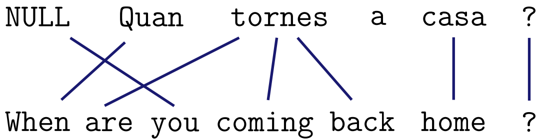
- Some sequences of words usually translate together.
- Approach: take sequences (phrases) as translation units.

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

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

SMT, components

The translation model $P(f|e)$

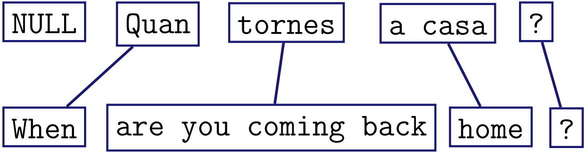


With the new translation units, $P(f|e)$ can be obtained following the same strategy as for word-based models with few modifications:

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

SMT, components

The translation model $P(f|e)$

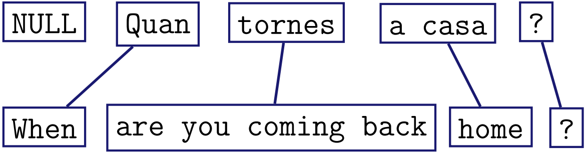


With the new translation units, $P(f|e)$ can be obtained following the same strategy as for word-based models with few modifications:

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

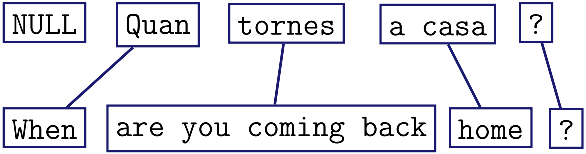
SMT, components

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 in phrases.
2. Translate each phrase into the target language.
3. Reorder the output.



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?

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.

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.

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.

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 $P(f|e)$

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:

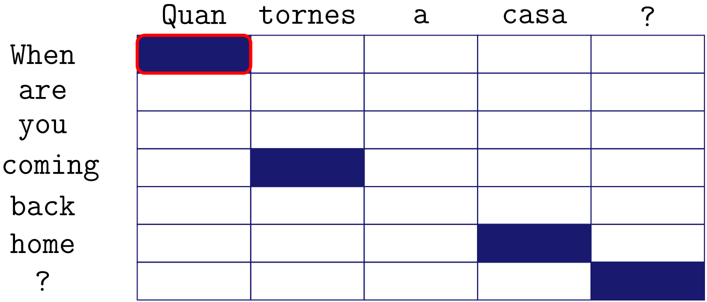
~~(Quan tornes, When are you coming back)~~

(Quan tornes tu, When are you coming back)

SMT, components

The translation model $P(f|e)$

Intersection

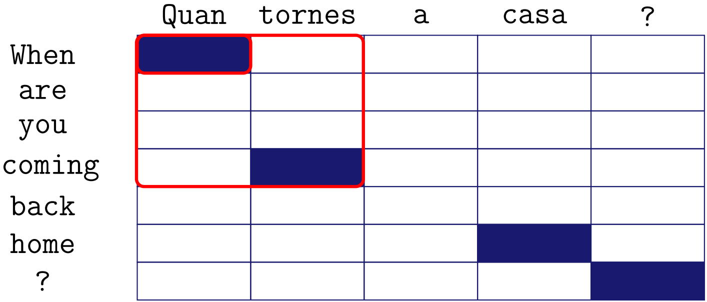


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

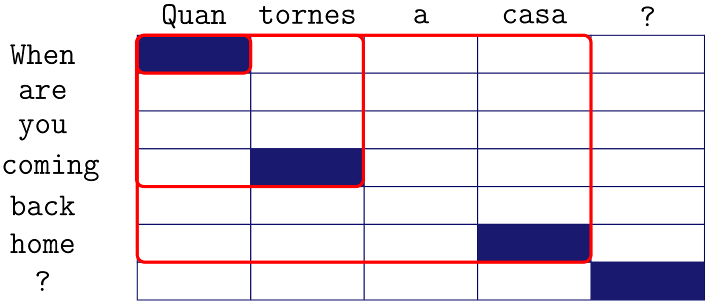


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

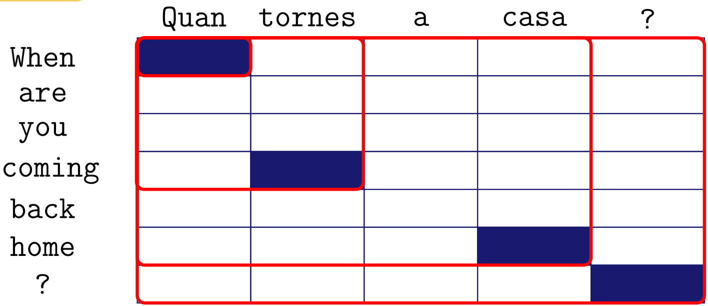


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

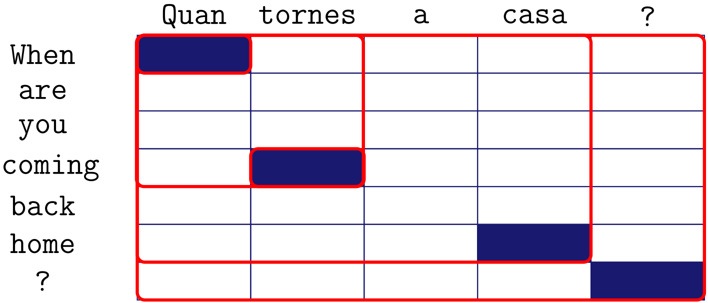


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

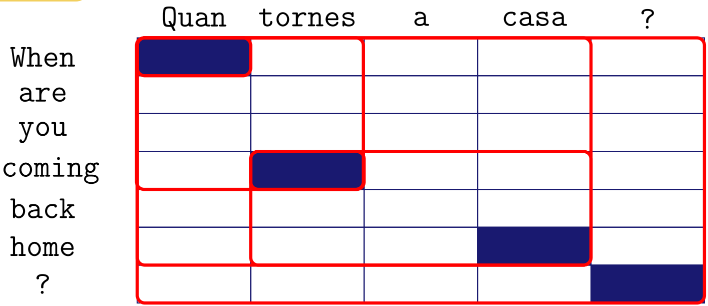


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

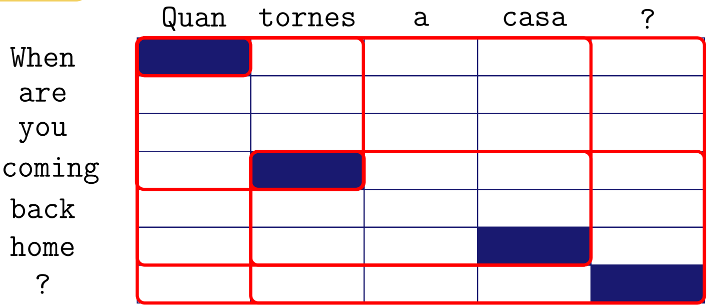


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

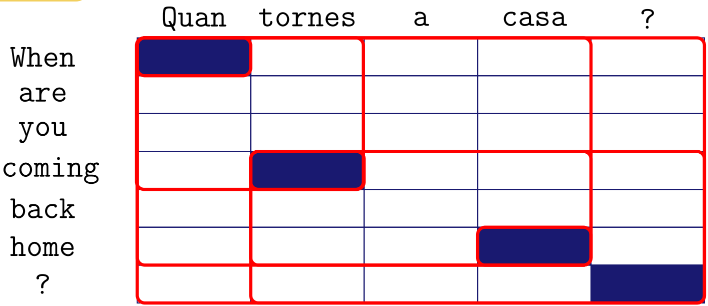


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

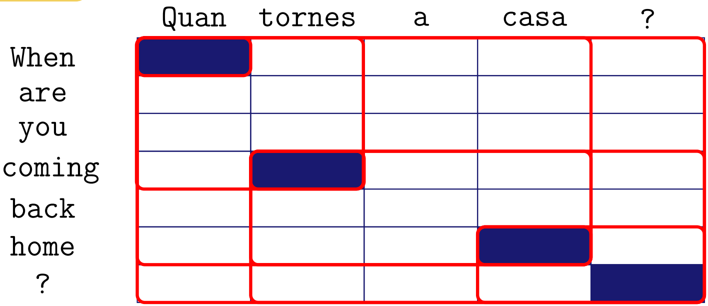


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

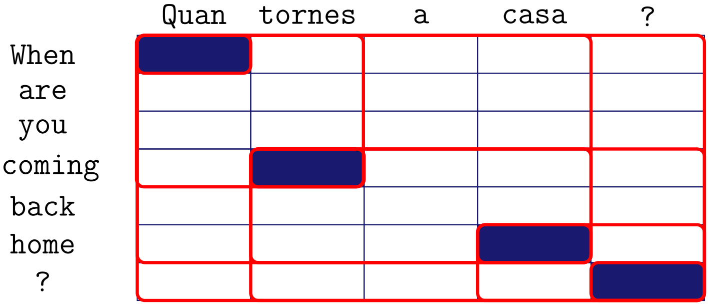


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Intersection

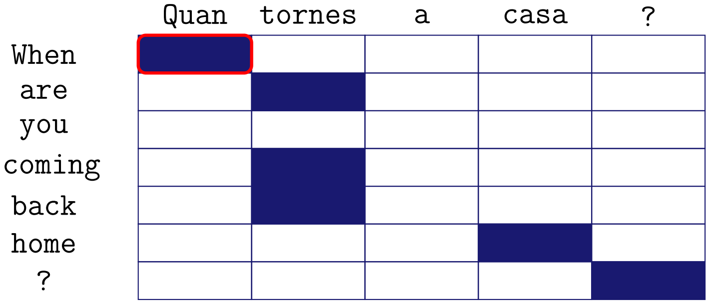


(Quan, When) (Quan tornes, When are you coming) (Quan tornes a casa, When are you coming back home) (Quan tornes a casa ?, When are you coming back home ?) (tornes, coming) (tornes a casa, coming back home) (tornes a casa ?, coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 10 phrases

SMT, components

The translation model $P(f|e)$

Union

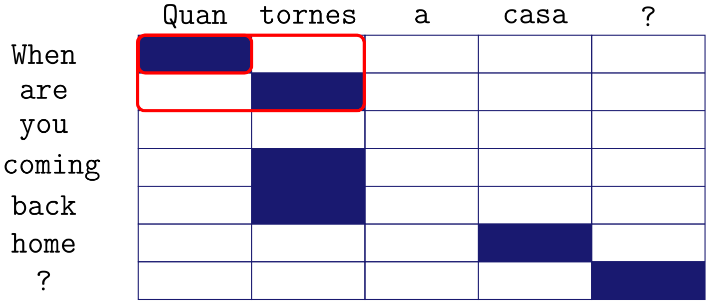


(Quan, When) (Quan tornes, When are) (Quan tornes, When are you coming) (Quan tornes, When are you coming back) (Quan tornes a casa, When are you coming back home) ... (tornes a casa ?, are you coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 21 phrases

SMT, components

The translation model $P(f|e)$

Union

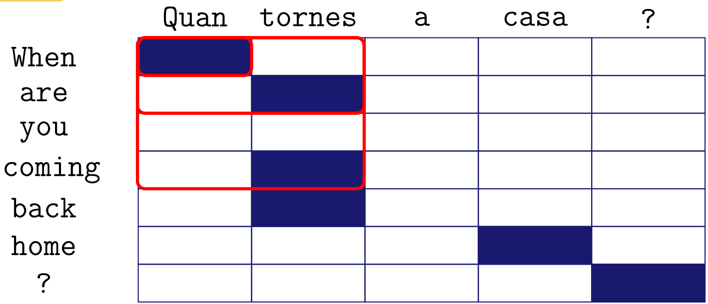


(Quan, When) (Quan tornes, When are) (Quan tornes, When are you coming) (Quan tornes, When are you coming back) (Quan tornes a casa, When are you coming back home) ... (tornes a casa ?, are you coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 21 phrases

SMT, components

The translation model $P(f|e)$

Union

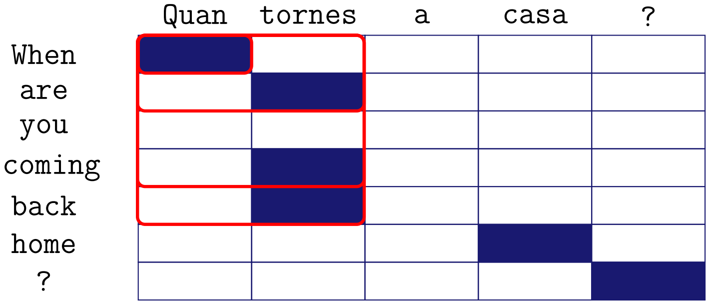


(Quan, When) (Quan tornes, When are) (Quan tornes, When are you coming) (Quan tornes, When are you coming back) (Quan tornes a casa, When are you coming back home) ... (tornes a casa ?, are you coming back home ?) (casa, home) (casa ?, home ?) (?, ?) 21 phrases

SMT, components

The translation model $P(f|e)$

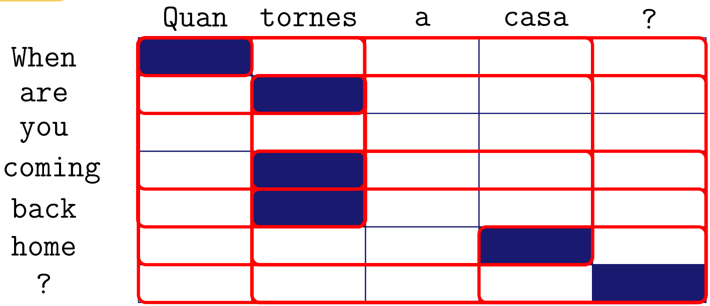
Union



(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.

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.

Decoder

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

Responsible for the search in the space of possible translations.

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

In our context, one can find:

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

Decoder

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

Responsible for the search in the space of possible translations.

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

In our context, one can find:

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

Decoder

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

Responsible for the search in the space of possible translations.

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

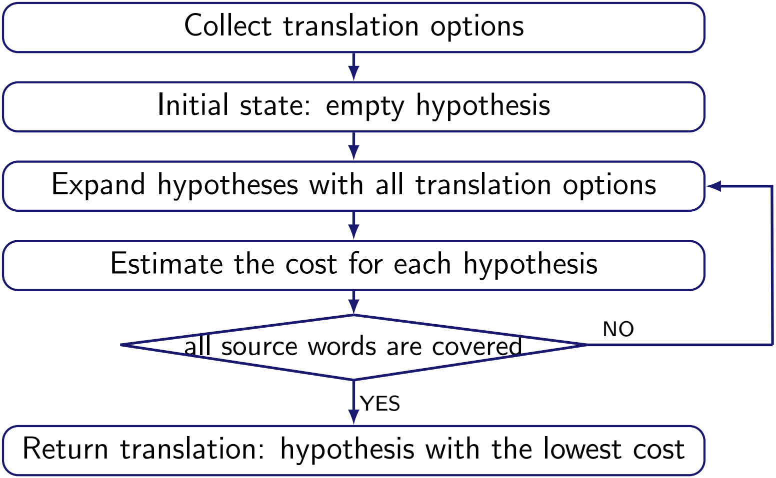
In our context, one can find:

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

SMT, components

A beam-search decoder

Core algorithm



Example: `Quan tornes a casa`

- Translation options:

  (`Quan`, `When`)
  (`Quan tornes`, `When are you coming back`)
  (`Quan tornes a casa`, `When are you coming back home`)
  (`tornes`, `come back`)
  (`tornes a casa`, `come back home`)
  (`a casa`, `home`)

Example: `Quan tornes a casa`

- Translation options:

  (Quan, When)
  (Quan tornes, When are you coming back)
  (Quan tornes a casa, When are you coming back home)
  (tornes, come back)
  (tornes a casa, come back home)
  (a casa, home)

- Notation for hypotheses in construction:

  Constructed sentence so far: come back
  Source words already translated: - x - -

SMT, components

A beam-search decoder

Example: `Quan tornes a casa`

- Translation options:

  ```
  (Quan, When)
  (Quan tornes, When are you coming back)
  (Quan tornes a casa, When are you coming back home)
  (tornes, come back)
  (tornes a casa, come back home)
  (a casa, home)
  ```

- Notation for hypotheses in construction:

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

Example: `Quan tornes a casa`

- Translation options:

  (`Quan`, `When`)
  (`Quan tornes`, `When are you coming back`)
  (`Quan tornes a casa`, `When are you coming back home`)
  (`tornes`, `come back`)
  (`tornes a casa`, `come back home`)
  (`a casa`, `home`)

- Initial hypothesis

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

SMT, components

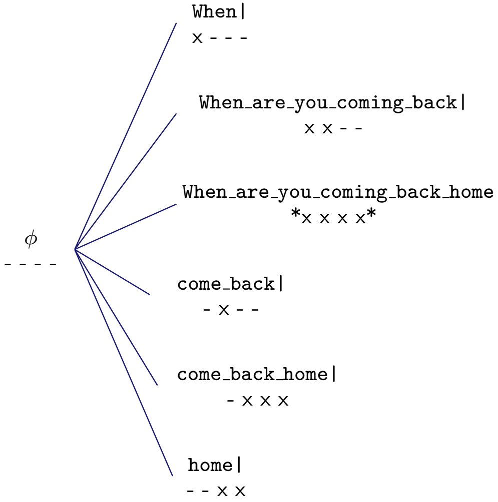
A beam-search decoder

ϕ

- - - -

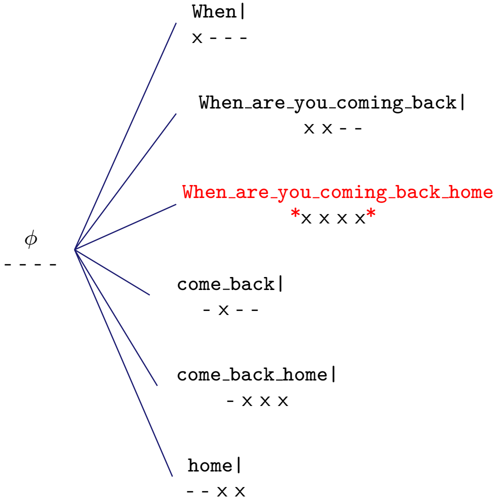
SMT, components

A beam-search decoder



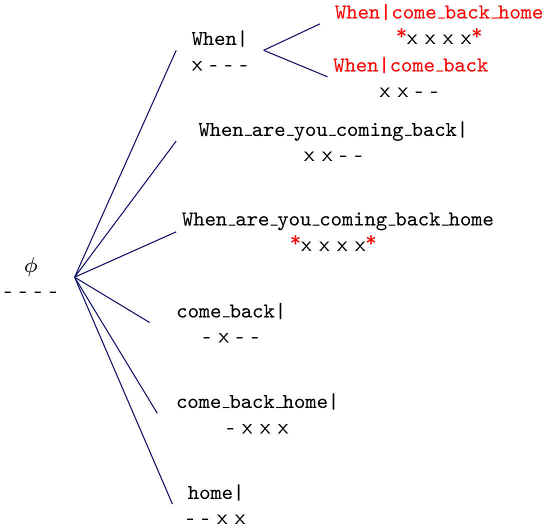
SMT, components

A beam-search decoder



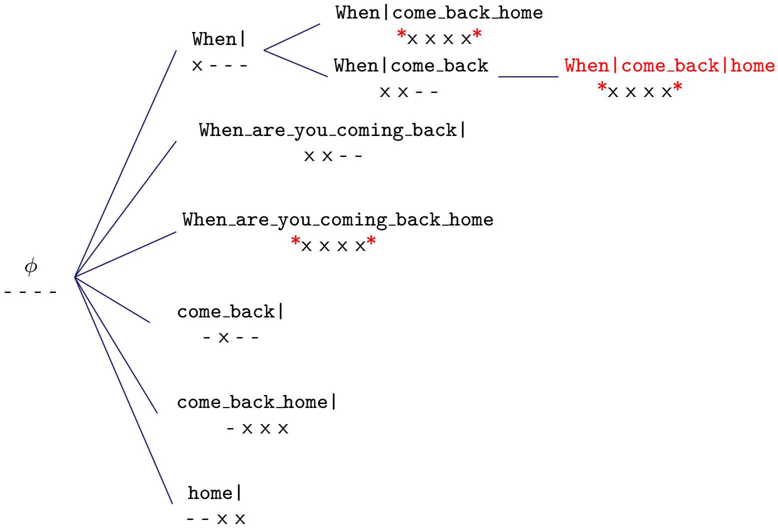
SMT, components

A beam-search decoder



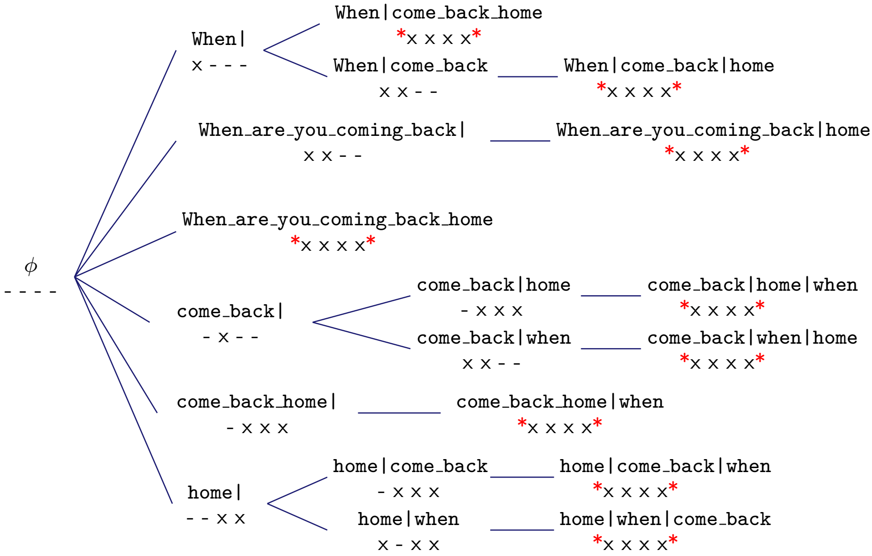
SMT, components

A beam-search decoder



SMT, components

A beam-search decoder



Exhaustive search

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

But...

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

Solution

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

Exhaustive search

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

But...

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

Solution

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

Exhaustive search

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

But...

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

Solution

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

Hypotheses recombination

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

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

x x x x   x x x x

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

Hypotheses recombination

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

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

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

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

Hypotheses recombination

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

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

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

Beam search and pruning (at last!)

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

What is an inferior hypothesis?

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

Beam search and pruning (at last!)

Strategy:

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

Decoding: keep in mind

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

Outline

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

Maximum likelihood (ML)

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

Maximum entropy (ME)

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

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

Log-linear model

SMT, the log-linear model
Motivation

Maximum likelihood (ML)

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

Maximum entropy (ME)

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

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

Log-linear model

SMT, the log-linear model

Motivation

Maximum likelihood (ML)

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

Maximum entropy (ME)

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

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

Log-linear model

SMT, the log-linear model

Motivation

Maximum likelihood (ML)

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

Maximum entropy (ME)

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

Log-linear model with

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

$\Rightarrow$ Maximum likelihood model

What can 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

State of the art 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.

State of the art 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.

State of the art 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:
  $P_d(\mathrm{ini}_{\mathrm{phrase}_i}, \mathrm{end}_{\mathrm{phrase}_{i-1}}) = \alpha|\mathrm{ini}_{\mathrm{phrase}_i} - \mathrm{end}_{\mathrm{phrase}_{i-1}} - 1|$

State of the art?

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

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

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

SMT, the log-linear model

Digression: lexicalised reordering or distortion

From where and how can one learn reorders?

(are, tornes, monotone)

SMT, the log-linear model

Digression: lexicalised reordering or distortion

From where and how can one learn reorders?

(coming back, tornes, swap)

SMT, the log-linear model

Digression: lexicalised reordering or distortion

From where and how can one learn reorders?

(home ?, casa ?, discontinuous)

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

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

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

State of the art feature functions

13 features may be used:

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

Development training, weights optimisation

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

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

Development training, weights optimisation

Strategies

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

Development training, weights optimisation

Strategies

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

Minimum Error-Rate Training

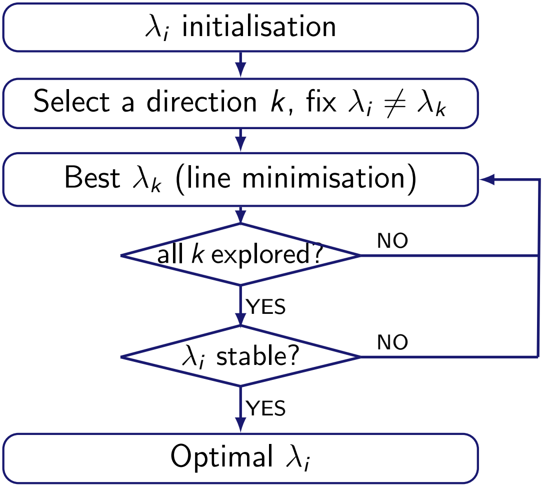
- Approach: Minimise an error function.

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

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

(More on MT Evaluation section)

Minimum Error-Rate Training rough algorithm



Log-linear model: keep in mind

- The log-linear model allows to include several weighted features. State of the art systems use 8 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 at most 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.

Outline

1 Introduction

2 Basics

3 Components

4 The log-linear model

5 Beyond standard SMT
- Factored translation models
- Syntactic translation models
- Ongoing research

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:

$$\begin{array}{cccccc}
\texttt{lemma}_f & \texttt{PoS}_f & \texttt{morphology}_f & & \texttt{word}_f \\
\downarrow \mathsf{T} & \downarrow \mathsf{T} & \downarrow \mathsf{T} & & \\
\texttt{lemma}_e & \texttt{PoS}_e & \texttt{morphology}_e & \xrightarrow{\mathsf{G}} & \texttt{word}_e
\end{array}$$

SMT, beyond standard SMT

Factored 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:

$$\begin{array}{ccccc} \texttt{casa}_f & \texttt{NN}_f & \texttt{fem., plural}_f & & \texttt{cases}_f \\ \downarrow \mathsf{T} & \downarrow \mathsf{T} & \downarrow \mathsf{T} & & \\ \texttt{house}_e & \texttt{NN}_e & \texttt{plural}_e & \xrightarrow{\mathsf{G}} & \texttt{houses}_e \end{array}$$

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.

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

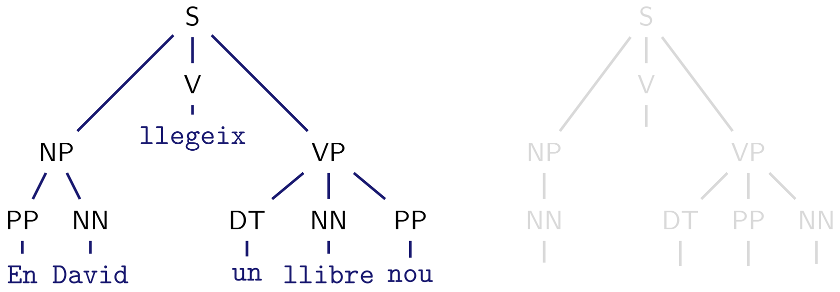
- Synchronous grammar formalism which learns a grammar that can simultaneously generate both trees.
  - Syntax-based. Respect linguistic units in translation.
  - Hierarchical phrase-based. Respect phrases in translation.

Syntactic models ease reordering. An intuitive example:

`En David llegeix un llibre nou`

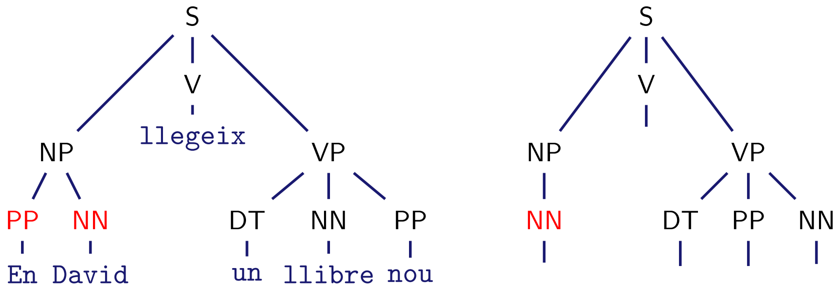
Syntactic models ease reordering. An intuitive example:

`En David llegeix un llibre nou`



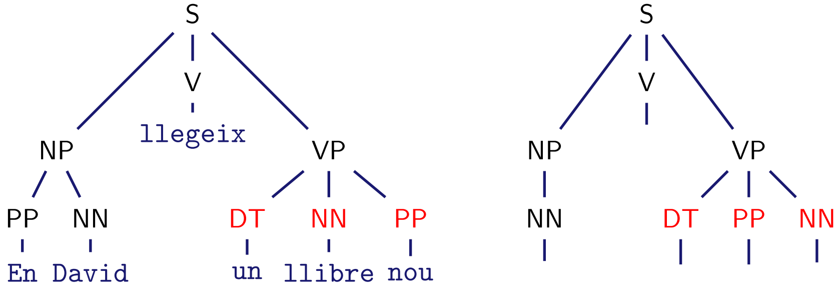
Syntactic models ease reordering. An intuitive example:

`En David llegeix un llibre nou`



Syntactic models ease reordering. An intuitive example:

`En David llegeix un llibre nou`

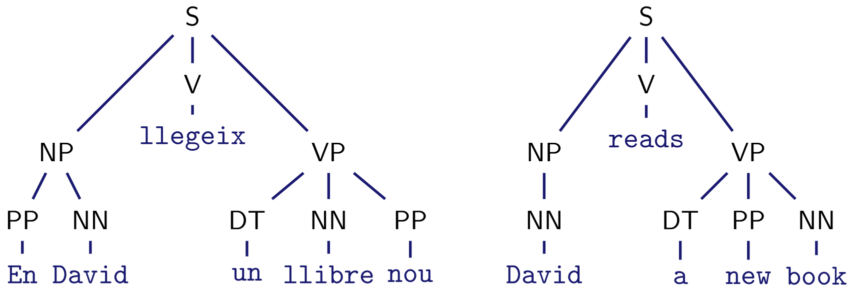


SMT, beyond standard SMT

Syntax-based translation models

Syntactic models ease reordering. An intuitive example:

`En David llegeix un llibre nou`



`David reads a new book`

Hot research topics

Current research on SMT addresses known and new problems.

Some components of the standard phrase-based model are still under study:

- Automatic alignments.
- Language models and smoothing techniques.
- Parameter optimisation.

Complements to a standard system can be added:

- Reordering as a pre-process or post-process.
- Reranking of n-best lists.
- OOV treatment.
- Domain adaptation.

SMT, beyond standard SMT

Ongoing research

Development of full systems from scratch or modifications to the standard:

- Using machine learning.
- Including linguistic information.
- Hybridation of MT paradigms.

Or a different strategy:

- Systems combination.

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.
- Current research addresses both new models and modifications to the existing ones.

Part II

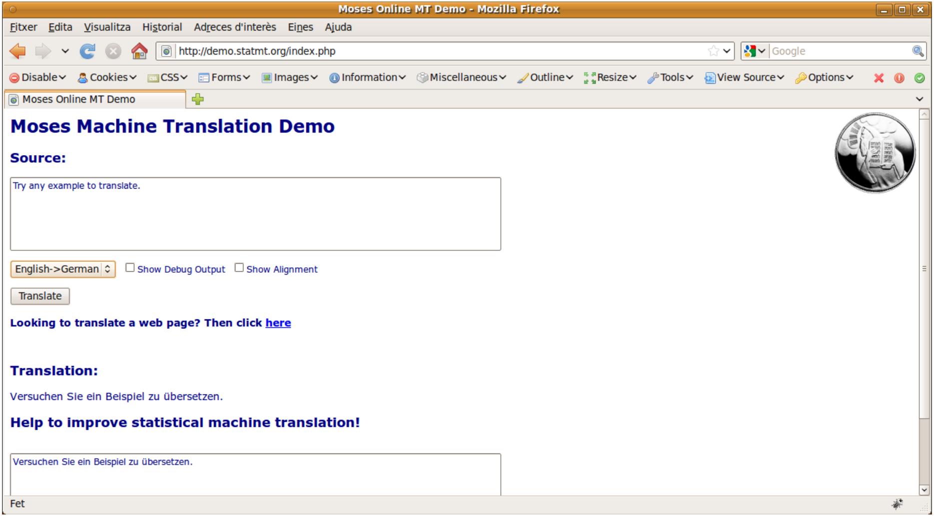
SMT experiments

Outline Part II

6. Translation system
    - Demos
    - Software
    - Steps

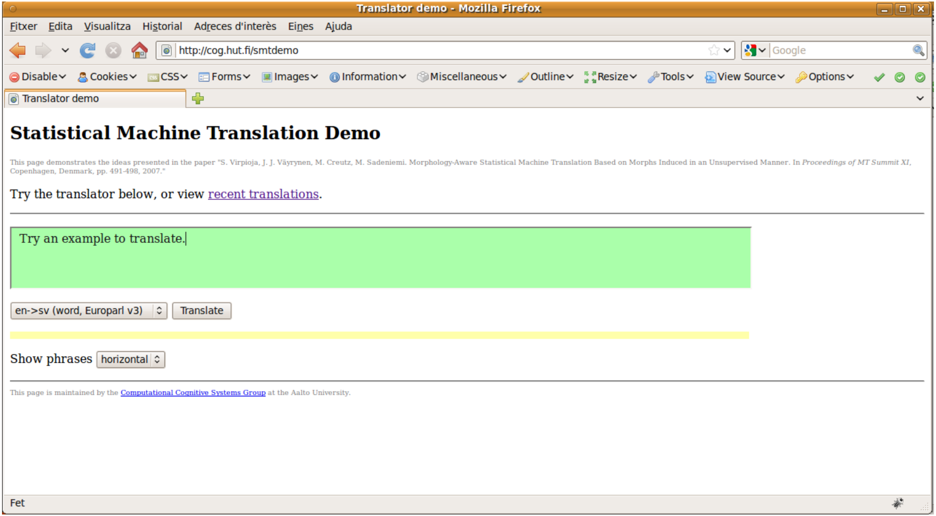
SMT system

Demo: http://demo.statmt.org/



SMT system

Demo: http://cog.hut.fi/smtdemo



Build your own SMT system

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

1. Download and prepare your data

1. Parallel corpora and some tools can be downloaded for instance from the WMT 2010 web page: http://www.statmt.org/wmt10/translation-task.html

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

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

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

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

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

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

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

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

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

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

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

2. Build the language model

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

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

3. Train the translation model

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

```
cristina@cosmos:~$ train-factored-phrase-model.perl -help
Train Phrase Model

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

Obre

3. Train the translation model (cont'd)

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

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

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

4. Tuning of parameters with MERT

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

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

2. Insert weights into configuration file with WMT10 script:

```
wmt10scripts/reuse-weights.perl ./tuning/moses.ini <
./model/moses.ini > moses.weight-reused.ini
```

SMT system

Steps

5. Run `Moses` decoder on a test set

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

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

3. Run the decoder:

```
moses/moses-cmd/src/moses -f ./filteredmodel/moses.ini <
testset.es > testset.translated.en
```

Part III

Machine Translation Evaluation

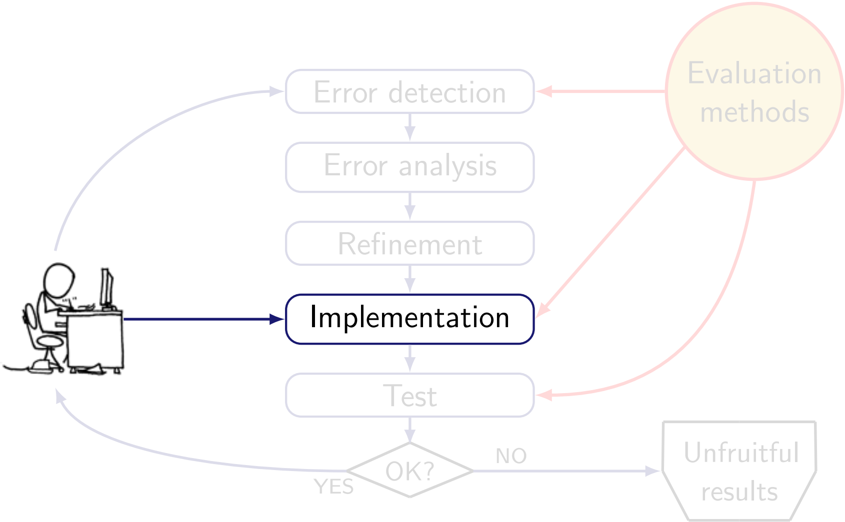
Outline

7. MT Evaluation basics
   - Automatic Evaluation
   - BLEU
   - Limits of lexical similarity

8. Evaluation system

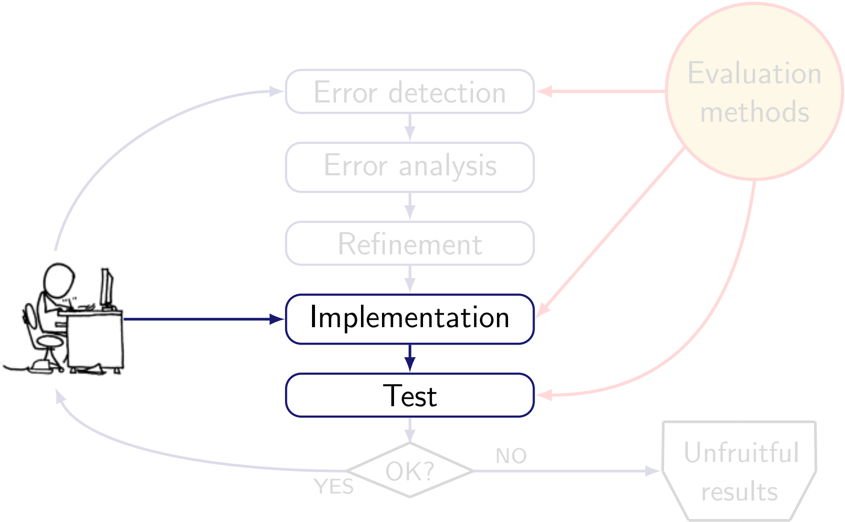
MT Evaluation

Importance for system development



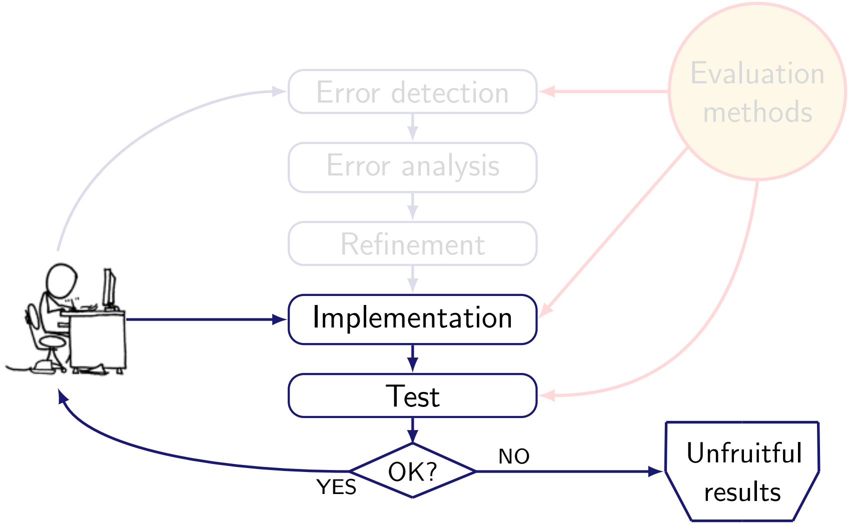
MT Evaluation

Importance for system development



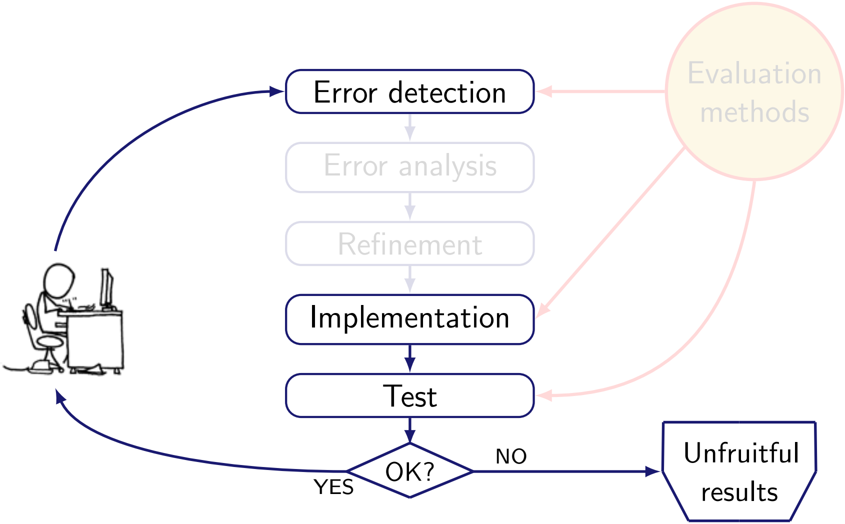
MT Evaluation

Importance for system development



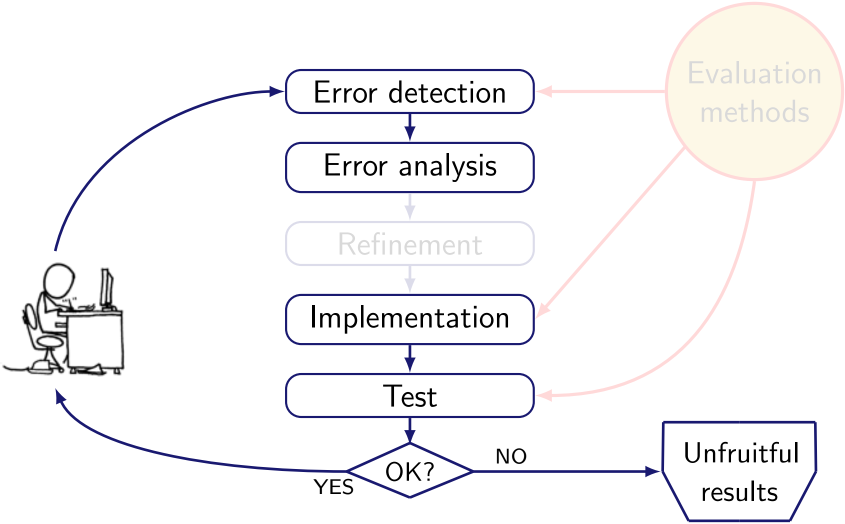
MT Evaluation

Importance for system development



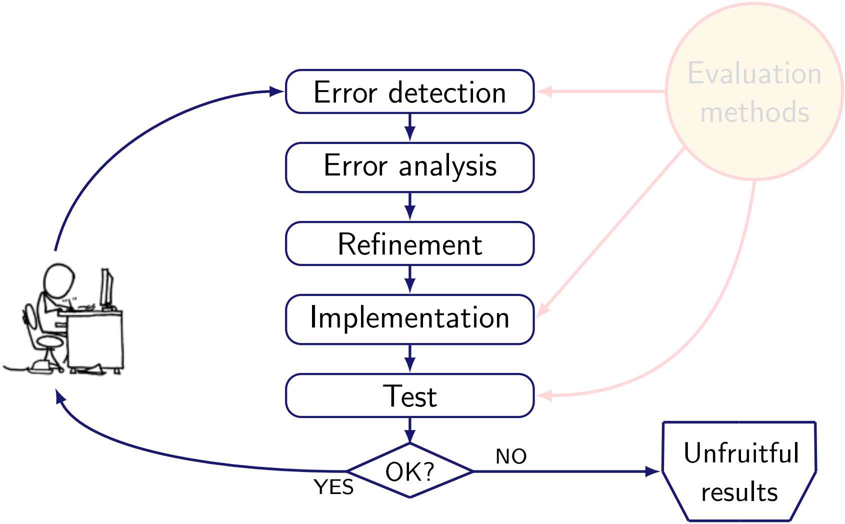
MT Evaluation

Importance for system development



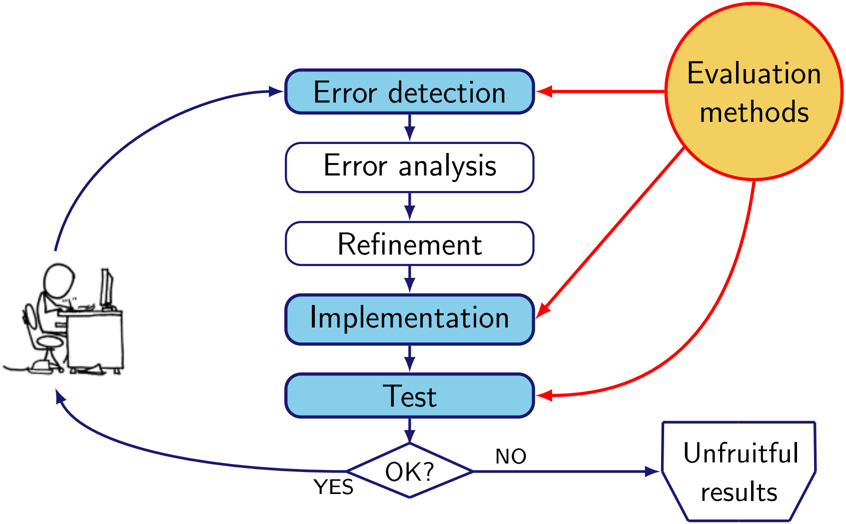
MT Evaluation

Importance for system development



MT Evaluation

Importance for system development



What can achieved with automatic evaluation
(as compared to manual evaluation)

- Automatic metrics notably accelerate the development cycle of MT systems:
  - Error analysis
  - System optimisation
  - System comparison

Besides, they are

- Costless (vs. costly)
- Objective (vs. subjective)
- Reusable (vs. non-reusable)

Metrics based on lexical similarity
(most of the metrics!)

- Edit Distance: WER, PER, TER
- Precision: BLEU, NIST, WNM
- Recall: ROUGE, CDER
- Precision/Recall: GTM, METEOR, BLANC, SIA

Metrics based on lexical similarity
(most of the metrics!)

- Edit Distance: WER, PER, TER
- Precision: BLEU, NIST, WNM
- Recall: ROUGE, CDER
- Precision/Recall: GTM, METEOR, BLANC, SIA

Nowadays, BLEU is accepted as *the standard* metric.

BLEU: a Method for Automatic Evaluation of Machine Translation

Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu
IBM Research Division

"The main idea is to use a weighted average of variable length phrase matches against the reference translations. This view gives rise to a family of metrics using various weighting schemes. We have selected a promising baseline metric from this family."

Candidate 1:

`It is a guide to action which ensures that the military always obeys the commands of the party.`

Candidate 2:

`It is to insure the troops forever hearing the activity guidebook that party direct.`

Candidate 1:

`It is a guide to action which ensures that the military always obeys the commands of the party.`

Reference 1:

`It is a guide to action that ensures that the military will forever heed Party commands.`

Reference 2:

`It is the guiding principle which guarantees the military forces always being under the command of the Party.`

Reference 3:

`It is the practical guide for the army always to heed the directions of the party.`

MT Evaluation

IBM BLEU: Papineni, Roukos, Ward and Zhu [2001]

Candidate 1:

`It is a guide to action which ensures that the military always obeys the commands of the party.`

Reference 1:

`It is a guide to action that ensures that the military will forever heed Party commands.`

Reference 2:

`It is the guiding principle which guarantees the military forces always being under the command of the Party.`

Reference 3:

`It is the practical guide for the army always to heed the directions of the party.`

Candidate 2:

`It is to insure the troops forever hearing the activity guidebook that party direct.`

Reference 1:

`It is a guide to action that ensures that the military will forever heed Party commands.`

Reference 2:

`It is the guiding principle which guarantees the military forces always being under the command of the Party.`

Reference 3:

`It is the practical guide for the army always to heed the directions of the party.`

Modified n-gram precision (1-gram)

Precision-based measure, but:

Candidate:

`The the the the the the the.`

Reference 1:

`The cat is on the mat.`

Reference 2:

`There is a cat on the mat.`

Modified n-gram precision (1-gram)

Precision-based measure, but: $\text{Prec.} = \frac{1 +}{7}$

Candidate:

`The the the the the the the.`

Reference 1:

`The cat is on the mat.`

Reference 2:

`There is a cat on the mat.`

Modified n-gram precision (1-gram)

Precision-based measure, but: $\text{Prec.} = \frac{2+}{7}$

Candidate:

`The the the the the the the.`

Reference 1:

`The cat is on the mat.`

Reference 2:

`There is a cat on the mat.`

Modified n-gram precision (1-gram)

Precision-based measure, but: $\text{Prec.} = \frac{3 +}{7}$

Candidate:

`The the the the the the the.`

Reference 1:

`The cat is on the mat.`

Reference 2:

`There is a cat on the mat.`

Modified n-gram precision (1-gram)

Precision-based measure, but: $\text{Prec.} = \frac{4 +}{7}$

Candidate:

`The the the the the the the.`

Reference 1:

`The cat is on the mat.`

Reference 2:

`There is a cat on the mat.`

Modified n-gram precision (1-gram)

Precision-based measure, but: $\text{Prec.} = \frac{5+}{7}$

Candidate:

`The the the the the the the.`

Reference 1:

`The cat is on the mat.`

Reference 2:

`There is a cat on the mat.`

Modified n-gram precision (1-gram)

Precision-based measure, but: $\text{Prec.} = \frac{6\,+}{7}$

Candidate:

`The the the the the the the.`

Reference 1:

`The cat is on the mat.`

Reference 2:

`There is a cat on the mat.`

Modified n-gram precision (1-gram)

Precision-based measure, but: $\text{Prec.} = \frac{7}{7}$

Candidate:

`The the the the the the the.`

Reference 1:

`The cat is on the mat.`

Reference 2:

`There is a cat on the mat.`

Modified n-gram precision (1-gram)

A reference word should only be matched once.

Algorithm:

1. Count number of times w_i occurs in each reference.
2. Keep the minimun between the maximum of (1) and the number of times w_i appears in the candidate (*clipping*).
3. Add these values and divide by candidate's number of words.

Modified n-gram precision (1-gram)

Modified 1-gram precision:

Candidate:

`The the the the the the the.`

Reference 1:

`The cat is on the mat.`

Reference 2:

`There is a cat on the mat.`

1. $w_i \rightarrow \mathrm{The}$
   $\#w_{i,R1} = 2$
   $\#w_{i,R2} = 1$
2. $\mathrm{Max}_{(1)}$=2, $\#w_{i,C} = 7$
   $\Rightarrow$ Min=2
3. No more distinct words

Modified n-gram precision (1-gram)

Modified 1-gram precision: $\mathrm{P}_1 =$

Candidate:

`The the the the the the the.`

Reference 1:

`The cat is on the mat.`

Reference 2:

`There is a cat on the mat.`

1. $w_i \to \mathrm{The}$
   $\#_{w_i,R1} = 2$
   $\#_{w_i,R2} = 1$
2. $\mathrm{Max}_{(1)}$=2, $\#_{w_i,C} = 7$
   $\Rightarrow$ Min=2
3. No more distinct words

Modified n-gram precision (1-gram)

Modified 1-gram precision: $P_1 = \frac{2}{}$

Candidate:

`The the the the the the the.`

Reference 1:

`The cat is on the mat.`

Reference 2:

`There is a cat on the mat.`

1. $w_i \rightarrow$ The
   $\#_{w_i,R1} = 2$
   $\#_{w_i,R2} = 1$
2. Max$_{(1)}$=2, $\#_{w_i,C} = 7$
   $\Rightarrow$ Min=2
3. No more distinct words

Modified n-gram precision (1-gram)

Modified 1-gram precision: $P_1 = \frac{2}{7}$

Candidate:

`The the the the the the the.`

Reference 1:

`The cat is on the mat.`

Reference 2:

`There is a cat on the mat.`

1. $w_i \rightarrow \mathrm{The}$
   $\#_{w_i,R1} = 2$
   $\#_{w_i,R2} = 1$
2. $\mathrm{Max}_{(1)}$=2, $\#_{w_i,C} = 7$
   $\Rightarrow$ Min=2
3. No more distinct words

Modified n-gram precision

- Straightforward generalisation to n-grams, P_n.
- Generalisation to multiple sentences:

$$\mathrm{P}_n = \frac{\sum_{C \in \{\text{candidates}\}} \sum_{n\text{gram} \in C} \mathit{Count}_{\text{clipped}}(n\text{gram})}{\sum_{C \in \{\text{candidates}\}} \sum_{n\text{gram} \in C} \mathit{Count}(n\text{gram})}$$

low n	high n
adequacy	fluency

Brevity penalty

Candidate:

`of the`

Reference 1:

`It is a guide to action that ensures that the military will forever heed Party commands.`

Reference 2:

`It is the guiding principle which guarantees the military forces always being under the command of the Party.`

Reference 3:

`It is the practical guide for the army always to heed the directions of the party.`

Brevity penalty

Candidate:

`of the`

$\mathrm{P}_1 = 2/2,\ \mathrm{P}_2 = 1/1$

Reference 1:

`It is a guide to action that ensures that the military will forever heed Party commands.`

Reference 2:

`It is the guiding principle which guarantees the military forces always being under the command of the Party.`

Reference 3:

`It is the practical guide for the army always to heed the directions of the party.`

Brevity penalty

$$\mathrm{BP} = \begin{cases} 1 & \text{if} \quad c > r \\ e^{1-r/c} & \text{if} \quad c \leq r \end{cases}$$

c candidate length, r reference length

- Multiplicative factor.
- At sentence level, huge punishment for short sentences.
- Estimated at document level.

BiLingual Evaluation Understudy, BLEU

$$\mathsf{BLEU} = \mathsf{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log \mathrm{P}_n\right)$$

- Geometric average of P_n (empirical suggestion).
- w_n positive weights summing to one.
- Brevity penalty.

Paper's Conclusions

- BLEU correlates with human judgements.
- It can distinguish among similar systems.
- Need for multiple references or a big test with heterogeneous references.
- More parametrisation in the future.

Watch out with BLEU implementations!

There are several widely used implementations of BLEU.

(`Moses multi-bleu.perl script, NIST mteval-vXX.pl script, etc.`)

Results differ because of:

- Different tokenisation approach.
- Different definition of *closest reference* in the brevity penalty estimation.

NIST is based on BLEU but:

- Arithmetic average of *n*-gram counts rather than a geometric average.
- Informative *n*-grams are given more weight.
- Different definition of brevity penalty.

Limits of lexical similarity

The reliability of lexical metrics depends very strongly on the heterogeneity/representativity of reference translations.

e: `This sentence is going to be difficult to evaluate.`

Ref1: `The evaluation of the clause is complicated.`
Ref2: `The sentence will be hard to qualify.`
Ref3: `The translation is going to be hard to evaluate.`
Ref4: `It will be difficult to punctuate the output.`

Lexical similarity is nor a sufficient neither a necessary condition so that two sentences convey the same meaning.

Limits of lexical similarity

The reliability of lexical metrics depends very strongly on the heterogeneity/representativity of reference translations.

e: `This sentence is going to be difficult to evaluate.`

Ref1: `The evaluation of the clause is complicated.`
Ref2: `The sentence will be hard to qualify.`
Ref3: `The translation is going to be hard to evaluate.`
Ref4: `It will be difficult to punctuate the output.`

Lexical similarity is nor a sufficient neither a necessary condition so that two sentences convey the same meaning.

Limits of lexical similarity

The reliability of lexical metrics depends very strongly on the heterogeneity/representativity of reference translations.

e: `This sentence is going to be difficult to evaluate.`

Ref1: `The evaluation of the clause is complicated.`
Ref2: `The sentence will be hard to qualify.`
Ref3: `The translation is going to be hard to evaluate.`
Ref4: `It will be difficult to punctuate the output.`

Lexical similarity is nor a sufficient neither a necessary condition so that two sentences convey the same meaning.

Recent efforts to go over lexical similarity

Extend the reference material:

- Using lexical variants such as morphological variations or synonymy lookup or using paraphrasing support.

Compare other linguistic features than words:

- Syntactic similarity: shallow parsing, full parsing (constituents /dependencies).
- Semantic similarity: named entities, semantic roles, discourse representations.

Combination of the existing metrics.

Recent efforts to go over lexical similarity

Extend the reference material:

- Using lexical variants such as morphological variations or synonymy lookup or using paraphrasing support.

Compare other linguistic features than words:

- Syntactic similarity: shallow parsing, full parsing (constituents /dependencies).
- Semantic similarity: named entities, semantic roles, discourse representations.

Combination of the existing metrics.

Recent efforts to go over lexical similarity

Extend the reference material:

- Using lexical variants such as morphological variations or synonymy lookup or using paraphrasing support.

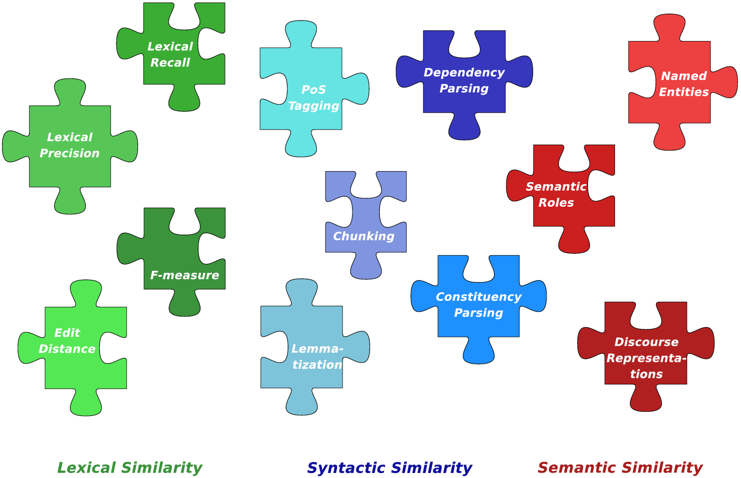
Compare other linguistic features than words:

- Syntactic similarity: shallow parsing, full parsing (constituents /dependencies).
- Semantic similarity: named entities, semantic roles, discourse representations.

Combination of the existing metrics.

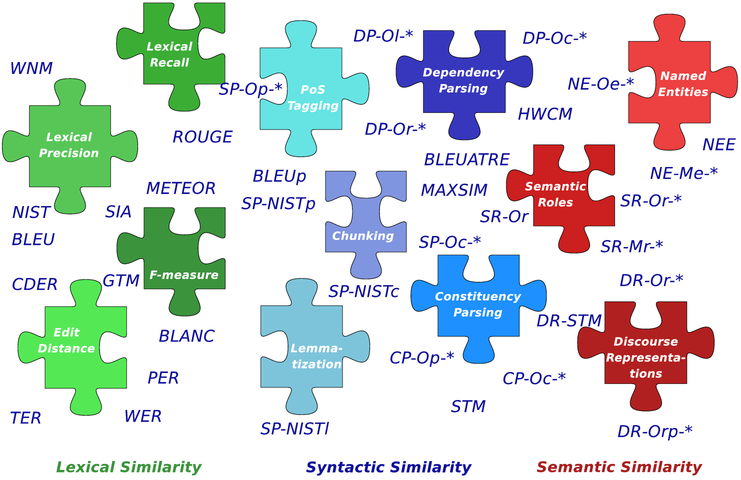
MT Evaluation

Towards Heterogeneous Automatic MT Evaluation



MT Evaluation

Towards Heterogeneous Automatic MT Evaluation



Asiya

Asiya has been designed to assist both **system** and metric **developers** by offering a rich repository of metrics and meta-metrics.

http://www.lsi.upc.edu/∼nlp/Asiya/

MT Evaluation: keep in mind

- Evaluation is important in the system development cycle. Automatic evaluation accelerates significatively the process.
- Up to now, most (common) metrics rely on lexical similarity, but it cannot assure a correct evaluation.
- Current work is being devoted to go beyond lexical similarity.

Outline

7 MT Evaluation basics

8 Evaluation system
- Software
- Steps
- Demo

Evaluate the results

1. With BLEU scoring tool. Available as a `Moses` script or from NIST:
   http://www.itl.nist.gov/iad/mig/tools/mtevalv13a-20091001.tar.gz
2. With `Asiya` package:
   http://www.lsi.upc.edu/∼nlp/Asiya/

1. Evaluate the results

1. With BLEU scoring tool in Moses:

```
moses/scripts/generic/multi-bleu.perl references.en <
testset.translated.en
```

1. With the Asiya toolkit:

```
Asiya.pl -eval single,ulc -g sys Asiya.config
```

```
input=raw

SRCLANG=de
TRGLANG=en
SRCCASE=cs
TRGCASE=cs

#SRC ==================================================
src=./data/patsA61P.test.de
#REF ==================================================
ref=./data/patsA61P.test.en
#OUT ==================================================
sys=./data/patsA61P.test.trans.de2en
sys=./data/patsA61P.test.trad.google.de2en
sys=./data/patsA61P.test.trad.bing.de2en
#-------------------------------------------------------
```

MT Evaluation

Steps

1. With the `Asiya` toolkit:

   `Asiya.pl -eval single,ulc -g sys Asiya.config`

```
input=raw

SRCLANG=de
TRGLANG=en
SRCCASE=cs
TRGCASE=cs

#SRC ==================================================
src=./data/patsA61P.test.de
#REF ==================================================
ref=./data/patsA61P.test.en
#OUT ===================================================
sys=./data/patsA61P.test.trans.de2en
sys=./data/patsA61P.test.trad.google.de2en
sys=./data/patsA61P.test.trad.bing.de2en
#-------------------------------------------------------
```

```
Asiya.pl -eval single,ulc -m metrSet Asiya.config
```

```
SRCLANG=de
TRGLANG=en

#SRC =====================================================
src=./data/patsA61P.test.de
#REF =====================================================
ref=./data/patsA61P.test.en
#OUT ======================================================
sys=./data/patsA61P.test.trans.de2en
#---------------------------------------------------------

metrSet=1-PER 1-TER 1-WER BLEU-4 CP-Oc-* CP-Op-* CP-STM-9 DP-HWC-c-4
DP-HWC-r-4 DP-HWC-w-4 DP-Oc-* DP-Ol-* DP-Or-* DR-Or-* DR-Orp-* DR-STM-9
GTM-1 GTM-2 GTM-3 MTR-exact MTR-stem MTR-wnstm MTR-wnsyn NE-Me-* NE-Oe-*
NE-Oe-** NIST-5 RG-L RG-S* RG-SU* RG-W-1.2 SP-Oc-* SP-Op-* SP-cNIST-5
SP-iobNIST-5 SP-lNIST-5 SP-pNIST-5 SR-Mr-* SR-Mrv-* SR-Or SR-Or-* SR-Orv
```

MT Evaluation

Metrics in Asiya (English)

```
------------------------------------------------------------------------------------------------------------------------
METRIC NAMES
------------------------------------------------------------------------------------------------------------------------
668 metrics are available for language 'en'

METRICS = { -PER, -TER, -TERbase, -TERp, -TERp-A, -WER, BLEU, BLEU-1, BLEU-2, BLEU-3, BLEU-4, BLEUi-2, BLEUi-3, BLEUi-4, CP-Oc(*), CP-Oc(ADJP), CP-Oc(ADVP), CP-Oc(CONJP), CP-Oc(FRA
G), CP-Oc(INTJ), CP-Oc(LST), CP-Oc(NAC), CP-Oc(NP), CP-Oc(NX), CP-Oc(O), CP-Oc(PP), CP-Oc(PRN), CP-Oc(PRT), CP-Oc(QP), CP-Oc(RRC), CP-Oc(S), CP-Oc(SBAR), CP-Oc(SINV), CP-Oc(SQ), CP
-Oc(UCP), CP-Oc(VP), CP-Oc(WHADJP), CP-Oc(WHADVP), CP-Oc(WHNP), CP-Oc(WHPP), CP-Oc(X), CP-Op(#), CP-Op($), CP-Op(''), CP-Op((), CP-Op()), CP-Op(*), CP-Op(,), CP-Op(.), CP-Op(:), CP
-Op(CC), CP-Op(CD), CP-Op(DT), CP-Op(EX), CP-Op(F), CP-Op(FW), CP-Op(IN), CP-Op(J), CP-Op(JJ), CP-Op(JJR), CP-Op(JJS), CP-Op(LS), CP-Op(MD), CP-Op(N), CP-Op(NN), CP-Op(NNP), CP-Op(
NNPS), CP-Op(NNS), CP-Op(P), CP-Op(PDT), CP-Op(POS), CP-Op(PRP$), CP-Op(PRP), CP-Op(R), CP-Op(RB), CP-Op(RBR), CP-Op(RBS), CP-Op(RP), CP-Op(SYM), CP-Op(TO), CP-Op(UH), CP-Op(V), CP
-Op(VB), CP-Op(VBD), CP-Op(VBG), CP-Op(VBN), CP-Op(VBP), CP-Op(VBZ), CP-Op(W), CP-Op(WDT), CP-Op(WP$), CP-Op(WP), CP-Op(WRB), CP-Op(``), CP-STM-1, CP-STM-2, CP-STM-3, CP-STM-4, CP-
STM-5, CP-STM-6, CP-STM-7, CP-STM-8, CP-STM-9, CP-STMi-2, CP-STMi-3, CP-STMi-4, CP-STMi-5, CP-STMi-6, CP-STMi-7, CP-STMi-8, CP-STMi-9, DP-HWCM_c-1, DP-HWCM_c-2, DP-HWCM_c-3, DP-HWC
M_c-4, DP-HWCM_r-1, DP-HWCM_r-2, DP-HWCM_r-3, DP-HWCM_r-4, DP-HWCM_w-1, DP-HWCM_w-2, DP-HWCM_w-3, DP-HWCM_w-4, DP-HWCMi_c-2, DP-HWCMi_c-3, DP-HWCMi_c-4, DP-HWCMi_r-2, DP-HWCMi_r-3,
 DP-HWCMi_r-4, DP-HWCMi_w-2, DP-HWCMi_w-3, DP-HWCMi_w-4, DP-Oc(*), DP-Oc(a), DP-Oc(as), DP-Oc(aux), DP-Oc(be), DP-Oc(c), DP-Oc(comp), DP-Oc(det), DP-Oc(have), DP-Oc(n), DP-Oc(postd
et), DP-Oc(ppspec), DP-Oc(predet), DP-Oc(prep), DP-Oc(saidx), DP-Oc(sentadjunct), DP-Oc(subj), DP-Oc(that), DP-Oc(u), DP-Oc(v), DP-Oc(vbe), DP-Oc(xsaid), DP-Ol(*), DP-Ol(1), DP-Ol(
2), DP-Ol(3), DP-Ol(4), DP-Ol(5), DP-Ol(6), DP-Ol(7), DP-Ol(8), DP-Ol(9), DP-Or(*), DP-Or(amod), DP-Or(amount-value), DP-Or(appo), DP-Or(appo-mod), DP-Or(as-arg), DP-Or(as1), DP-Or
(as2), DP-Or(aux), DP-Or(be), DP-Or(being), DP-Or(by-subj), DP-Or(c), DP-Or(cn), DP-Or(comp1), DP-Or(conj), DP-Or(desc), DP-Or(dest), DP-Or(det), DP-Or(else), DP-Or(fc), DP-Or(gen)
, DP-Or(guest), DP-Or(have), DP-Or(head), DP-Or(i), DP-Or(inv-aux), DP-Or(inv-have), DP-Or(lex-dep), DP-Or(lex-mod), DP-Or(mod), DP-Or(mod-before), DP-Or(neg), DP-Or(nn), DP-Or(num
), DP-Or(num-mod), DP-Or(obj), DP-Or(obj1), DP-Or(obj2), DP-Or(p), DP-Or(p-spec), DP-Or(pcomp-c), DP-Or(pcomp-n), DP-Or(person), DP-Or(pnmod), DP-Or(poss), DP-Or(post), DP-Or(pre),
 DP-Or(pred), DP-Or(punc), DP-Or(rel), DP-Or(s), DP-Or(sc), DP-Or(subcat), DP-Or(subclass), DP-Or(subj), DP-Or(title), DP-Or(vrel), DP-Or(wha), DP-Or(whn), DP-Or(whp), DPm-HWCM_c-1
, DPm-HWCM_c-2, DPm-HWCM_c-3, DPm-HWCM_c-4, DPm-HWCM_r-1, DPm-HWCM_r-2, DPm-HWCM_r-3, DPm-HWCM_r-4, DPm-HWCM_w-1, DPm-HWCM_w-2, DPm-HWCM_w-3, DPm-HWCM_w-4, DPm-HWCMi_c-2, DPm-HWCMi
_c-3, DPm-HWCMi_c-4, DPm-HWCMi_r-2, DPm-HWCMi_r-3, DPm-HWCMi_r-4, DPm-HWCMi_w-2, DPm-HWCMi_w-3, DPm-HWCMi_w-4, DPm-Oc(*), DPm-Oc(......), DPm-Ol(*), DPm-Ol(1), DPm-Ol(2), DPm-Ol(3)
, DPm-Ol(4), DPm-Ol(5), DPm-Ol(6), DPm-Ol(7), DPm-Ol(8), DPm-Ol(9), DPm-Or(*), DPm-Or(.......), DR-Fr(*), DR-Frp(*), DR-Ol, DR-Or(*), DR-Or(*)_b, DR-Or(*)_i, DR-Or(alfa), DR-Or(car
d), DR-Or(drs), DR-Or(eq), DR-Or(imp), DR-Or(merge), DR-Or(named), DR-Or(not), DR-Or(or), DR-Or(pred), DR-Or(prop), DR-Or(rel), DR-Or(smerge), DR-Or(timex), DR-Or(whq), DR-Or-(dr),
 DR-Orp(*), DR-Orp(*)_b, DR-Orp(*)_i, DR-Orp(alfa), DR-Orp(card), DR-Orp(dr), DR-Orp(drs), DR-Orp(eq), DR-Orp(imp), DR-Orp(merge), DR-Orp(named), DR-Orp(not), DR-Orp(or), DR-Orp(pr
ed), DR-Orp(prop), DR-Orp(rel), DR-Orp(smerge), DR-Orp(timex), DR-Orp(whq), DR-Pr(*), DR-Prp(*), DR-Rr(*), DR-Rrp(*), DR-STM-1, DR-STM-2, DR-STM-3, DR-STM-4, DR-STM-4_b, DR-STM-4_i
, DR-STM-5, DR-STM-6, DR-STM-7, DR-STM-8, DR-STM-9, DR-STMi-2, DR-STMi-3, DR-STMi-4, DR-STMi-5, DR-STMi-6, DR-STMi-7, DR-STMi-8, DR-STMi-9, DRdoc-Ol, DRdoc-Or(*), DRdoc-Or(*)_b, DR
doc-Or(*)_i, DRdoc-Or(alfa), DRdoc-Or(card), DRdoc-Or(dr), DRdoc-Or(drs), DRdoc-Or(eq), DRdoc-Or(imp), DRdoc-Or(merge), DRdoc-Or(named), DRdoc-Or(not), DRdoc-Or(or), DRdoc-Or(pred)
, DRdoc-Or(prop), DRdoc-Or(rel), DRdoc-Or(smerge), DRdoc-Or(timex), DRdoc-Or(whq), DRdoc-Orp(*), DRdoc-Orp(*)_b, DRdoc-Orp(*)_i, DRdoc-Orp(alfa), DRdoc-Orp(card), DRdoc-Orp(dr), DR
doc-Orp(drs), DRdoc-Orp(eq), DRdoc-Orp(imp), DRdoc-Orp(merge), DRdoc-Orp(named), DRdoc-Orp(not), DRdoc-Orp(or), DRdoc-Orp(pred), DRdoc-Orp(prop), DRdoc-Orp(rel), DRdoc-Orp(smerge),
 DRdoc-Orp(timex), DRdoc-Orp(whq), DRdoc-STM-1, DRdoc-STM-2, DRdoc-STM-3, DRdoc-STM-4, DRdoc-STM-4_b, DRdoc-STM-4_i, DRdoc-STM-5, DRdoc-STM-6, DRdoc-STM-7, DRdoc-STM-8, DRdoc-STM-9
, DRdoc-STMi-2, DRdoc-STMi-3, DRdoc-STMi-4, DRdoc-STMi-5, DRdoc-STMi-6, DRdoc-STMi-7, DRdoc-STMi-8, DRdoc-STMi-9, Fl, GTM-1, GTM-2, GTM-3, METEOR-ex, METEOR-pa, METEOR-st, METEOR-s
y, NE-Me(*), NE-Me(ANGLE_QUANTITY), NE-Me(DATE), NE-Me(DISTANCE_QUANTITY), NE-Me(LANGUAGE), NE-Me(LOC), NE-Me(MEASURE), NE-Me(METHOD), NE-Me(MISC), NE-Me(MONEY), NE-Me(NUM), NE-Me(
ORG), NE-Me(PER), NE-Me(PERCENT), NE-Me(PROJECT), NE-Me(SIZE_QUANTITY), NE-Me(SPEED_QUANTITY), NE-Me(SYSTEM), NE-Me(TEMPERATURE_QUANTITY), NE-Me(TIME), NE-Me(WEIGHT_QUANTITY), NE-O
e(*), NE-Oe(**), NE-Oe(ANGLE_QUANTITY), NE-Oe(DATE), NE-Oe(DISTANCE_QUANTITY), NE-Oe(LANGUAGE), NE-Oe(LOC), NE-Oe(MEASURE), NE-Oe(METHOD), NE-Oe(MISC), NE-Oe(MONEY), NE-Oe(NUM), NE
-Oe(O), NE-Oe(ORG), NE-Oe(PER), NE-Oe(PERCENT), NE-Oe(PROJECT), NE-Oe(SIZE_QUANTITY), NE-Oe(SPEED_QUANTITY), NE-Oe(SYSTEM), NE-Oe(TEMPERATURE_QUANTITY), NE-Oe(TIME), NE-Oe(WEIGHT_Q
UANTITY), NIST, NIST-1, NIST-2, NIST-3, NIST-4, NIST-5, NISTi-2, NISTi-3, NISTi-4, NISTi-5, Ol, Pl, ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, ROUGE-L, ROUGE-S*, ROUGE-SU*, ROUGE-W, Rl, S
P-Oc(*), SP-Oc(ADJP), SP-Oc(ADVP), SP-Oc(CONJP), SP-Oc(INTJ), SP-Oc(LST), SP-Oc(NP), SP-Oc(O), SP-Oc(PP), SP-Oc(PRT), SP-Oc(SBAR), SP-Oc(UCP), SP-Oc(VP), SP-Op(#), SP-Op($), SP-Op(
''), SP-Op((), SP-Op()), SP-Op(*), SP-Op(,), SP-Op(.), SP-Op(:), SP-Op(CC), SP-Op(CD), SP-Op(DT), SP-Op(EX), SP-Op(F), SP-Op(FW), SP-Op(IN), SP-Op(J), SP-Op(JJ), SP-Op(JJR), SP-Op(
JJS), SP-Op(LS), SP-Op(MD), SP-Op(N), SP-Op(NN), SP-Op(NNP), SP-Op(NNPS), SP-Op(NNS), SP-Op(P), SP-Op(PDT), SP-Op(POS), SP-Op(PRP$), SP-Op(PRP), SP-Op(R), SP-Op(RB), SP-Op(RBR), SP
-Op(RBS), SP-Op(RP), SP-Op(SYM), SP-Op(TO), SP-Op(UH), SP-Op(V), SP-Op(VB), SP-Op(VBD), SP-Op(VBG), SP-Op(VBN), SP-Op(VBP), SP-Op(VBZ), SP-Op(W), SP-Op(WDT), SP-Op(WP$), SP-Op(WP),
 SP-Op(WRB), SP-Op(``), SP-cNIST, SP-cNIST-1, SP-cNIST-2, SP-cNIST-3, SP-cNIST-4, SP-cNIST-5, SP-cNISTi-2, SP-cNISTi-3, SP-cNISTi-4, SP-cNISTi-5, SP-iobNIST, SP-iobNIST-1, SP-iobNI
ST-2, SP-iobNIST-3, SP-iobNIST-4, SP-iobNIST-5, SP-iobNISTi-2, SP-iobNISTi-3, SP-iobNISTi-4, SP-iobNISTi-5, SP-lNIST, SP-lNIST-1, SP-lNIST-2, SP-lNIST-3, SP-lNIST-4, SP-lNIST-5, SP
-lNISTi-2, SP-lNISTi-3, SP-lNISTi-4, SP-lNISTi-5, SP-pNIST, SP-pNIST-1, SP-pNIST-2, SP-pNIST-3, SP-pNIST-4, SP-pNIST-5, SP-pNISTi-2, SP-pNISTi-3, SP-pNISTi-4, SP-pNISTi-5, SR-Fr(*)
, SR-MFr(*), SR-MPr(*), SR-MRr(*), SR-Mr(*), SR-Mr(*)_b, SR-Mr(*)_i, SR-Mr(A0), SR-Mr(A1), SR-Mr(A2), SR-Mr(A3), SR-Mr(A4), SR-Mr(A5), SR-Mr(AA), SR-Mr(AM-ADV), SR-Mr(AM-CAU), SR-M
r(AM-DIR), SR-Mr(AM-DIS), SR-Mr(AM-EXT), SR-Mr(AM-LOC), SR-Mr(AM-MNR), SR-Mr(AM-MOD), SR-Mr(AM-NEG), SR-Mr(AM-PNC), SR-Mr(AM-PRD), SR-Mr(AM-REC), SR-Mr(AM-TMP), SR-Mrv(*), SR-Mrv(*
)_b, SR-Mrv(*)_i, SR-Mrv(A0), SR-Mrv(A1), SR-Mrv(A2), SR-Mrv(A3), SR-Mrv(A4), SR-Mrv(A5), SR-Mrv(AA), SR-Mrv(AM-ADV), SR-Mrv(AM-CAU), SR-Mrv(AM-DIR), SR-Mrv(AM-DIS), SR-Mrv(AM-EXT)
, SR-Mrv(AM-LOC), SR-Mrv(AM-MNR), SR-Mrv(AM-MOD), SR-Mrv(AM-NEG), SR-Mrv(AM-PNC), SR-Mrv(AM-PRD), SR-Mrv(AM-REC), SR-Mrv(AM-TMP), SR-Nv, SR-Ol, SR-Or, SR-Or(*), SR-Or(*)_b, SR-Or(*
)_i, SR-Or(A0), SR-Or(A1), SR-Or(A2), SR-Or(A3), SR-Or(A4), SR-Or(A5), SR-Or(AA), SR-Or(AM-ADV), SR-Or(AM-CAU), SR-Or(AM-DIR), SR-Or(AM-DIS), SR-Or(AM-EXT), SR-Or(AM-LOC), SR-Or(AM
-MNR), SR-Or(AM-MOD), SR-Or(AM-NEG), SR-Or(AM-PNC), SR-Or(AM-PRD), SR-Or(AM-REC), SR-Or(AM-TMP), SR-Or_b, SR-Or_i, SR-Orv, SR-Orv(*), SR-Orv(*)_b, SR-Orv(*)_i, SR-Orv(A0), SR-Orv(A
1), SR-Orv(A2), SR-Orv(A3), SR-Orv(A4), SR-Orv(A5), SR-Orv(AA), SR-Orv(AM-ADV), SR-Orv(AM-CAU), SR-Orv(AM-DIR), SR-Orv(AM-DIS), SR-Orv(AM-EXT), SR-Orv(AM-LOC), SR-Orv(AM-MNR), SR-O
rv(AM-MOD), SR-Orv(AM-NEG), SR-Orv(AM-PNC), SR-Orv(AM-PRD), SR-Orv(AM-REC), SR-Orv(AM-TMP), SR-Orv_b, SR-Orv_i, SR-Ov, SR-Pr(*), SR-Rr(*) }
```

2. Evaluate the results on-line

1. OpenMT Evaluation Demo
   http://biniki.lsi.upc.edu/openMT/evaldemo.php

MT Evaluation

Demo: http://biniki.lsi.upc.edu/openMT/evaldemo.php

OpenMT Evaluation Demo - Mozilla Firefox

Fitxer Edita Visualitza Historial Adreces d'interès Eines Ajuda

http://biniki.lsi.upc.edu/openMT/evaldemo.php

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

OpenMT Evaluation Demo

OpenMT Evaluation Demo

Linguistic Features towards Hetereogeneous Automatic MT Evaluation

Translation quality aspects are heterogeneous and diverse, involving, in general, many different linguistic dimensions. However, most automatic evaluation methods in use today rely on partial quality assumptions, such as lexical similarity. This introduces a bias in the development cycle which in some cases has been reported to carry very negative consequences. In order to tackle this methodological problem, we explore a novel path towards heterogeneous automatic MT evaluation. We have compiled a rich set of specialized similarity metrics operating at different linguistic levels (lexical, syntactic and semantic). We have also studied how the scores conferred by different metrics may be integrated into a single measure of quality, without having to adjust their relative importance.

This demo allows you to obtain automatic evaluation scores according to a selected set of metric representatives, together with ULC combined score (i.e., arithmetic mean) over a heuristically defined set of metrics.

Instructions:

- Select the target language. The metric set will depend on this choice. Currently, linguistic features are only supported for English. For other languages, the metric set limits to the lexical dimension.
- Type test cases in the text areas below:
  - at least one, and up to five, candidate translations.
  - at least one, and up to five, reference translations.
  - text areas must contain one test case per line up to a maximum of ten.
- Click on the "Go Evaluate!" button at the right.
  - Execution should take between one and five minutes for English, and a just a few seconds for other languages.

Target Language English

Go Evaluate!

Fet

Part IV

Appendix: References

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]

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]

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]

References

Language model

- Kneser & Ney, 1995 [KN95]

MERT

- Och, 2003 [Och03]

Domain adaptation

- Bertoldi and Federico, 2009 [Och03]

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]

References

Alternative systems in development

- Blunsom et al., 2008 [BCO08]
- Canisius & van den Bosch, 2009 [CvdB09]
- Chiang et al., 2009 [CKW09]
- Finch & Sumita, 2009 [FS09]
- Hassan et al., 2009 [HSW09]
- Shen et al., 2009 [SXZ+09]

References

Evaluation

- Papineni, 2002 [PRWZ02]
- Doddington, 2002 [Dod02]
- Banerjee & Alon Lavie, 2005 [BL05]
- Giménez & Amigó, 2006 [GA06]

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]

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.

Phil Blunsom, Trevor Cohn, and Miles Osborne.
A discriminative latent variable model for statistical machine translation.
In *ACL-08: HLT. 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 200–208, 2008.

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.

References II

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.

Satanjeev Banerjee and Alon Lavie.
METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments.
In *Proceedings of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, 2005.

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.

References III

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.

David Chiang, Kevin Knight, and Wei Wang.
11,001 new features for statistical machine translation.
In *NAACL '09: Human Language Technologies: the 2009 annual conference of the North American Chapter of the ACL*, pages 218–226. Association for Computational Linguistics, 2009.

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

Sander Canisius and Antal van den Bosch.
A constraint satisfaction approach to machine translation.
In Lluís Màrquez and Harold Somers, editors, *EAMT-2009: Proceedings of the 13th Annual Conference of the European Association for Machine Translation*, pages 182–189, 2009.

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.

George Doddington.
Automatic evaluation of machine translation quality using n-gram co-occurrence statistics.
In *Proceedings of the 2nd Internation Conference on Human Language Technology*, pages 138–145, 2002.

References V

Andrew Finch and Eiichiro Sumita.
Bidirectional phrase-based statistical machine translation.
In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1124–1132, Singapore, August 2009. Association for Computational Linguistics.

Jesús Giménez and Enrique Amigó.
IQMT: A Framework for Automatic Machine Translation Evaluation.
In *Proceedings of the 5th LREC*, pages 685–690, 2006.

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

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

Hany Hassan, Khalil Sima'an, and Andy Way.
A syntactified direct translation model with linear-time decoding.
In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1182–1191, Singapore, August 2009. Association for Computational Linguistics.

References VI

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.

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.

References VII

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.

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.

References VIII

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu.
Bleu: a method for automatic evaluation of machine translation.
In *Proceedings of the Association of Computational Linguistics*, pages 311–318, 2002.

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

References IX

Libin Shen, Jinxi Xu, Bing Zhang, Spyros Matsoukas, and Ralph Weischedel.
Effective use of linguistic and contextual information for statistical machine translation.
In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 72–80, Singapore, August 2009. Association for Computational Linguistics.

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.