Statistical Machine Translation
A practical tutorial

Cristina España i Bonet
LSI Department
Universitat Politècnica de Catalunya

MOLTO Kickoff Meeting
UPC, Barcelona
11th March, 2010
Overview

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

Part I: SMT background

~ 90 minutes
Overview

7 Translation system

8 Evaluation system

Part II: SMT experiments

∼ 30 minutes

9 References

Part III: References
Part I

SMT background
Introduction

Machine Translation Taxonomy

Machine Translation systems

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

Machine Translation Taxonomy

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

Machine Translation systems

Human Translation with Machine Support

Machine Translation with Human Support

Fully Automated Translation

MOLTO’s core

Empirical systems

Rule-based systems
Introduction

Machine Translation Taxonomy

Machine Translation systems

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

Empirical systems

- Statistical Machine Translation
- Example-based Translation

Rule-based systems
Introduction

Machine Translation Taxonomy

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

MOLTO’s extension
Empirical MT relies on large parallel aligned corpora.

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.
Empirical MT relies on large parallel aligned corpora.

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.
## Introduction

Empirical Machine Translation

### Aligned parallel corpora numbers

<table>
<thead>
<tr>
<th>Corpus</th>
<th># segments (app.)</th>
<th># words (app.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JRC-Acquis</td>
<td>$1.0 \cdot 10^6$</td>
<td>$30 \cdot 10^6$</td>
</tr>
<tr>
<td>Europarl</td>
<td>$1.5 \cdot 10^6$</td>
<td>$45 \cdot 10^6$</td>
</tr>
<tr>
<td>United Nations</td>
<td>$3.8 \cdot 10^6$</td>
<td>$100 \cdot 10^6$</td>
</tr>
</tbody>
</table>

### Books

<table>
<thead>
<tr>
<th>Title</th>
<th># words (approx.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Bible</td>
<td>$0.8 \cdot 10^6$</td>
</tr>
<tr>
<td>The Dark Tower series</td>
<td>$1.2 \cdot 10^6$</td>
</tr>
<tr>
<td>Encyclopaedia Britannica</td>
<td>$44 \cdot 10^6$</td>
</tr>
</tbody>
</table>
## Introduction

Empirical Machine Translation

### Aligned parallel corpora numbers

<table>
<thead>
<tr>
<th>Corpus</th>
<th># segments (app.)</th>
<th># words (app.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JRC-Acquis</td>
<td>$1.0 \cdot 10^6$</td>
<td>$30 \cdot 10^6$</td>
</tr>
<tr>
<td>Europarl</td>
<td>$1.5 \cdot 10^6$</td>
<td>$45 \cdot 10^6$</td>
</tr>
<tr>
<td>United Nations</td>
<td>$3.8 \cdot 10^6$</td>
<td>$100 \cdot 10^6$</td>
</tr>
</tbody>
</table>

### Books

<table>
<thead>
<tr>
<th>Title</th>
<th># words (approx.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Bible</td>
<td>$0.8 \cdot 10^6$</td>
</tr>
<tr>
<td>The Dark Tower series</td>
<td>$1.2 \cdot 10^6$</td>
</tr>
<tr>
<td>Encyclopaedia Britannica</td>
<td>$44 \cdot 10^6$</td>
</tr>
</tbody>
</table>
Outline

1 Introduction
2 Basics
3 Components
4 The log-linear model
5 Beyond standard SMT
6 MT Evaluation
SMT, basics

The beginnings, summarised timeline

- **1950**: ENIAC
- **1949**: Weaver memo
- **1966**: Alpac report
- **1988**: Candide
- **2001**: BLEU
- **2000**: MT systems (Empirical vs. Rule-based)
SMT, basics
The beginnings, summarised timeline

- 1946: ENIAC
- 1949: Weaver memo
- 1966: Alpac report
- 1988: Candide

Timeline:
- 1950
- 1975
- 2000
SMT, basics
The beginnings, summarised timeline

- 1946: ENIAC
- 1949: Weaver memo
- 1966: Alpac report
- 1988: Candide

1950: Dictionary MT systems
1975: Rule-based MT systems
2000:
SMT, basics
The beginnings, summarised timeline

1946 ENIAC
1949 Weaver memo
1966 Alpac report
1988 Candide
2001 BLEU

1950
Dictionary MT systems
1975
Rule-based MT systems
2000
Empirical MT systems
The Noisy Channel as a statistical approach to translation:

Good morning!
The Noisy Channel as a statistical approach to translation:

Good morning! → Bon jour!
The Noisy Channel as a statistical approach to translation:

e: Good morning!

f: Bon jour!

Language $E \ (e \in E)$

Language $F \ (f \in F)$

translation
SMT, basics
The Noisy Channel approach

Mathematically:

\[
P(e|f) = \frac{P(e) P(f|e)}{P(f)}
\]

\[
T(f) = \hat{e} = \arg\max_e P(e|f) = \arg\max_e P(e) P(f|e)
\]
**SMT, basics**

**Components**

\[
T(f) = \hat{e} = \arg\max_e P(e) P(f|e)
\]

**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

\[ T(f) = \hat{e} = \arg\max_e P(e) P(f|e) \]

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

\arg\max

- Search done by the decoder
SMT, basics

Components

\[ T(f) = \hat{e} = \arg\max_e P(e) P(f|e) \]

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

\textbf{argmax}
- Search done by the \textit{decoder}
Outline

1. Introduction
2. Basics
3. Components
   - Language model
   - Translation model
   - Decoder
4. The log-linear model
5. Beyond standard SMT
6. MT Evaluation
SMT, components
The language model $P(e)$

Language model

$$T(f) = \hat{e} = \arg\max_e P(e) P(f|e)$$

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.
  $\implies$ Long sentences may have zero probability!
The language model $P(e)$

**Language model**

$$T(f) = \hat{e} = \arg\max_e P(e) P(f|e)$$

Estimation of how probable a sentence is.

Naïve estimation on a corpus with $N$ sentences:

Frequentist probability of a sentence $e$:

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

Problem:

- Long chains are difficult to observe in corpora.
  - Long sentences may have zero probability!
**Language model**

\[ T(f) = \hat{e} = \text{argmax}_e P(e) P(f|e) \]

Estimation of how probable a sentence is.

Naïve estimation on a corpus with \( N \) sentences:

Frequentist probability of a sentence \( e \):

\[ P(e) = \frac{N_e}{N_{\text{sentences}}} \]

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 $e \Rightarrow \{w_1, \ldots, w_m\}$.

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

- 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

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

where, for each factor,

\[
P(\text{and}|\phi, \text{All}, \text{work}) = \frac{N(\text{All work and})}{N(\text{All work})}
\]
Example, a 4-gram model

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

\[
P(e) = P(\text{All}|\phi, \phi, \phi) \cdot P(\text{work}|\phi, \phi, \text{All}) \cdot P(\text{and}|\phi, \text{All}, \text{work}) \cdot P(\text{no}|\text{All}, \text{work}, \text{and}) \cdot P(\text{play}|\text{work}, \text{and}, \text{no}) \cdot P(\text{makes}|\text{and}, \text{no}, \text{play}) \cdot P(\text{Jack}|\text{no}, \text{play}, \text{makes}) \cdot P(\text{a}|\text{play}, \text{makes}, \text{Jack}) \cdot P(\text{dull}|\text{makes}, \text{Jack}, \text{a}) \cdot P(\text{boy}|\text{Jack}, \text{a}, \text{dull})
\]

where, for each factor,

\[
P(\text{and}|\phi, \text{All}, \text{work}) = \frac{N(\text{All work and})}{N(\text{All work})}
\]
Example, a 4-gram model

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

\[ P(e) = P(\text{All}|\phi,\phi,\phi) \cdot P(\text{work}|\phi,\phi,\text{All}) \cdot P(\text{and}|\phi,\text{All},\text{work}) \]
\[ \cdot P(\text{no}|\text{All,work,and}) \cdot P(\text{play}|\text{work,and,}\text{no}) \]
\[ \cdot P(\text{makes}|\text{and,}\text{no,play}) \cdot P(\text{Jack}|\text{no,play,}\text{makes}) \]
\[ \cdot P(\text{a}|\text{play,}\text{makes,Jack}) \cdot P(\text{dull}|\text{makes,Jack,a}) \]
\[ \cdot P(\text{boy}|\text{Jack,a,dull}) \]

where, for each factor,

\[ P(\text{and}|\phi,\text{All,work}) = \frac{N(\text{All work and})}{N(\text{All work})} \]
SMT, components
The language model $P(e)$

Example, a 4-gram model

$$e: \text{All work and no play makes Jack a dull boy}$$

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

where, for each factor,

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

The language model $P(e)$

Example, a 4-gram model

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

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

where, for each factor,

$$P(\text{and}|\phi, \text{All}, \text{work}) = \frac{N(\text{All work and})}{N(\text{All work})}$$
Example, a 4-gram model

\[ e: \text{All work and no play makes Jack a dull boy} \]

\[
P(e) = P(\text{All}|\phi, \phi, \phi) \cdot P(\text{work}|\phi, \phi, \text{All}) \cdot P(\text{and}|\phi, \text{All}, \text{work}) \cdot \]
\[ P(\text{no}|\text{All}, \text{work}, \text{and}) \cdot P(\text{play}|\text{work}, \text{and}, \text{no}) \cdot \]
\[ P(\text{makes}|\text{and}, \text{no}, \text{play}) \cdot P(\text{Jack}|\text{no}, \text{play}, \text{makes}) \cdot \]
\[ P(\text{a}|\text{play}, \text{makes}, \text{Jack}) \cdot P(\text{dull}|\text{makes}, \text{Jack}, \text{a}) \cdot \]
\[ P(\text{boy}|\text{Jack}, \text{a}, \text{dull}) \]

where, for each factor,

\[
P(\text{and}|\phi, \text{All}, \text{work}) = \frac{N(\text{All work and})}{N(\text{All work})} \]
SMT, components

The language model $P(e)$

Example, a 4-gram model

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

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

where, for each factor,

$$P(\text{and}|\phi, All, work) = \frac{N(\text{All work and})}{N(\text{All work})}$$
Example, a 4-gram model

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

\[ P(e) = P(\text{All} | \phi, \phi, \phi) P(\text{work} | \phi, \phi, \text{All}) P(\text{and} | \phi, \text{All}, \text{work}) \]
\[ \quad P(\text{no} | \text{All}, \text{work}, \text{and}) P(\text{play} | \text{work}, \text{and}, \text{no}) \]
\[ \quad P(\text{makes} | \text{and}, \text{no}, \text{play}) P(\text{Jack} | \text{no}, \text{play}, \text{makes}) \]
\[ \quad P(\text{a} | \text{play}, \text{makes}, \text{Jack}) P(\text{dull} | \text{makes}, \text{Jack}, \text{a}) \]
\[ \quad P(\text{boy} | \text{Jack}, \text{a}, \text{dull}) \]

where, for each factor,

\[ P(\text{and} | \phi, \text{All}, \text{work}) = \frac{N(\text{All work and})}{N(\text{All work})} \]
Example, a 4-gram model

\[ e: \text{All work and no play makes Jack a dull boy} \]

\[
P(e) = P(\text{All} | \phi, \phi, \phi) \cdot P(\text{work} | \phi, \phi, \text{All}) \cdot P(\text{and} | \phi, \text{All}, \text{work}) \cdot P(\text{no} | \text{All}, \text{work}, \text{and}) \cdot P(\text{play} | \text{work}, \text{and}, \text{no}) \cdot P(\text{makes} | \text{and}, \text{no}, \text{play}) \cdot P(\text{Jack} | \text{no}, \text{play}, \text{makes}) \cdot P(\text{a} | \text{play}, \text{makes}, \text{Jack}) \cdot P(\text{dull} | \text{makes}, \text{Jack}, \text{a}) \cdot P(\text{boy} | \text{Jack}, \text{a}, \text{dull})
\]

where, for each factor,

\[
P(\text{and} | \phi, \text{All}, \text{work}) = \frac{N(\text{All work and})}{N(\text{All work})}
\]
SMT, components

The language model $P(e)$

Example, a 4-gram model

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

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

where, for each factor,

$$P(\text{and}|\phi, \text{All}, \text{work}) = \frac{N(\text{All work and})}{N(\text{All work})}$$
Example, a 4-gram model

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

\[ P(e) = P(\text{All}|\phi, \phi, \phi) \cdot P(\text{work}|\phi, \phi, \text{All}) \cdot P(\text{and}|\phi, \text{All}, \text{work}) \]
\[ P(\text{no}|\text{All, work, and}) \cdot P(\text{play}|\text{work, and, no}) \]
\[ P(\text{makes}|\text{and, no, play}) \cdot P(\text{Jack}|\text{no, play, makes}) \]
\[ P(\text{a}|\text{play, makes, Jack}) \cdot P(\text{dull}|\text{makes, Jack, a}) \]
\[ P(\text{boy}|\text{Jack, a, dull}) \]

where, for each factor,
\[ P(\text{and}|\phi, \text{All, work}) = \frac{N(\text{All work and})}{N(\text{All work})} \]
Example, a 4-gram model

\[ P(e) = P(\text{All} | \phi, \phi, \phi) \cdot P(\text{work} | \phi, \phi, \text{All}) \cdot P(\text{and} | \phi, \text{All}, \text{work}) \]
\[ P(\text{no} | \text{All}, \text{work}, \text{and}) \cdot P(\text{play} | \text{work}, \text{and}, \text{no}) \]
\[ P(\text{makes} | \text{and}, \text{no}, \text{play}) \cdot P(\text{Jack} | \text{no}, \text{play}, \text{makes}) \]
\[ P(\text{a} | \text{play}, \text{makes}, \text{Jack}) \cdot P(\text{dull} | \text{makes}, \text{Jack}, \text{a}) \]
\[ P(\text{boy} | \text{Jack}, \text{a}, \text{dull}) \]

where, for each factor,

\[ P(\text{and} | \phi, \text{All}, \text{work}) = \frac{N(\text{All work and})}{N(\text{All work})} \]
Main problems and criticisms:

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

Solution

Smoothing techniques:

- Linear interpolation.

\[
P(\text{and}|\text{All, work}) = \frac{N(\text{All, work, and})}{N(\text{All, work})} + \lambda_2 \frac{N(\text{work, and})}{N(\text{work})} + \lambda_1 \frac{N(\text{and})}{N(\text{words})} + \lambda_0
\]
Main problems and criticisms:

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

Solution

Smoothing techniques:

- Linear interpolation.
- Back-off models.

\[
P(\text{and}|\text{All, work}) = \frac{N(\text{All, work, and})}{N(\text{All, work})} + \lambda_2 \frac{N(\text{work, and})}{N(\text{work})} + \lambda_1 \frac{N(\text{and})}{N(\text{words})} + \lambda_0
\]
Main problems and criticisms:

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

Solution

Smoothing techniques:

- Linear interpolation.

$$P(\text{and} | \text{All, work}) = \frac{N_{(\text{All, work, and})}}{N_{(\text{All, work})}} + \lambda_2 \frac{N_{(\text{work, and})}}{N_{(\text{work})}} + \lambda_1 \frac{N_{(\text{and})}}{N_{\text{words}}} + \lambda_0$$
Main problems and criticisms:

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

Solution

Smoothing techniques:

- Linear interpolation.

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

$$T(f) = \hat{e} = \arg\max_e P(e) P(f|e)$$

Estimation of the lexical correspondence between languages.

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

NULL Quan tornes a casa ?

When are you coming back home ?
SMT, components
The translation model $P(f|e)$

Translation model

$$T(f) = \hat{e} = \arg\max_e P(e) P(f|e)$$

Estimation of the lexical correspondence between languages.

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

NULL  Quan tornes a casa ?

When are you coming back home ?
SMT, components
The translation model $P(f|e)$

**Translation model**

$$T(f) = \hat{e} = \arg\max_e P(e) P(f|e)$$

Estimation of the lexical correspondence between languages.

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

```
NULL    Quan    tornes    a    casa    ?
When    are    you    coming    back    home    ?
```
One should at least model for each word in the source language:

- Its translation,
- the number of necessary words in the target language,
- the position of the translation within the sentence,
- and, besides, the number of words that need to be generated from scratch.
SMT, components
The translation model $P(f|e)$

**Word-based models: the IBM models**

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

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

- **Fertility** $n$
  
  $n(3|\text{tornes})$: 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 $p_1$**
  
  $p(you|NULL)$: the prob. that the spurious word *you* is generated (from *NULL*).
SMT, components
The translation model $P(f|e)$

Back to the example:

NULL Quan tornes a casa ?

NULL Quan tornes tornes tornes tornes casa ?

NULL When are coming back home ?

you When are coming back home ?

When are you coming back home ?

Fertility
Translation
Insertion
Distortion
SMT, components
The translation model \( P(f|e) \)

Back to the example:

```
NULL  Quan tornes a casa ?
NULL  Quan tornestornestornes casa ?
NULL  When are coming back home ?
you   When are coming back home ?
   X  When are you coming back home ?
```

- **Fertility**
- **Translation**
- **Insertion**
- **Distortion**
SMT, components
The translation model $P(f|e)$

Back to the example:

NULL Quan tornes a casa ?

NULL Quantornestornestornes casa ?

NULL When are coming back home ?

you When are coming back home ?

When are you coming back home ?

Fertility
Translation
Insertion
Distortion
SMT, components

The translation model $P(f|e)$

Back to the example:

NULL Quan tornes a casa ?

NULL Quan tornestornestornes casa ?

NULL When are coming back home ?

you When are coming back home ?

When are you coming back home ?
SMT, components
The translation model $P(f|e)$

Back to the example:

```
NULL Quan tornes a casa ?
```

```
NULL Quan tornes tornes tornes tornes casa ?
```

```
NULL When are coming back home ?
```

```
you When are coming back home ?
```

```
When are you coming back home ?
```

Fertility
Translation
Insertion
Distortion
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 million sentences word by word.
- Estimate word alignments together with the parameters.
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 million 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

Converged?

NO \hspace{2cm} YES

Final parameters and alignments
SMT, components

The translation model $P(f|e)$

**Expectation-Maximisation algorithm**

1. Parameter initialisation
2. Alignment probability calculation
3. Parameter reestimation
4. Alignment probability recalculation

**Converged?**

NO

YES

Final parameters and alignments
SMT, components
The translation model $P(f|e)$

**Expectation-Maximisation algorithm**

Parameter initialisation →
Alignment probability calculation →
Parameter reestimation →
Alignment probability recalculation →
Converged?

- NO →
- YES →

Final parameters and alignments
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

**NULL** Quan tornes a casa ?

When are you coming back home ?

English to Catalan

**NULL** When are you coming back home ?

Quan tornes a casa ?
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

NULL Quan tornes a casa?

When are you coming back home?

English to Catalan

NULL When are you coming back home?

Quan tornes a casa?
SMT, components
The translation model $P(f|e)$

Graphically:

<table>
<thead>
<tr>
<th></th>
<th>NULL</th>
<th>Quan</th>
<th>tornes</th>
<th>a</th>
<th>casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>When</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>are</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>you</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>coming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>back</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>home</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Graphically:

<table>
<thead>
<tr>
<th>NULL</th>
<th>Quan</th>
<th>tornes</th>
<th>a</th>
<th>casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>When</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>are</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>you</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>coming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>back</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>home</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

English to Catalan
**SMT, components**

The translation model $P(f|e)$

Alignment symmetrisation

- **Intersection**: high-confidence, **high precision**.

<table>
<thead>
<tr>
<th></th>
<th>NULL</th>
<th>Quan</th>
<th>tornes</th>
<th>a</th>
<th>casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>When</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>are</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>you</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>coming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>back</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>home</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Catalan to English $\cap$ 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

\text{f: En} \text{ David llegeix el llibre nou.}
\text{e: } \emptyset
From Word-based to Phrase-based models

f: En David llegeix el llibre nou.

e: David
The translation model $P(f|e)$

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

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

From Word-based to Phrase-based models

f: En David llegeix el llibre nou.
e: David reads the new book. ✓

f: \textcolor{red}{En} David llegeix el llibre de nou.
e: \emptyset
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 again.
SMT, components
The translation model $P(f|e)$

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 book again.
SMT, components
The translation model \( P(f|e) \)

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

From Word-based to Phrase-based models

f: En David llegaix el llibre nou.
e: David reads the new book. ✓

f: En David llegaix 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. ✓

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

From Word-based to Phrase-based models

\[ f: \text{En David llegeix el llibre nou.} \]
\[ e: \text{David reads the new book.} \quad \checkmark \]

\[ f: \text{En David llegeix el llibre de nou.} \]
\[ e: \text{David reads the book of new.} \quad \times \]
\[ e: \text{David reads the} \]
SMT, components
The translation model $P(f|e)$

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. X
e: David reads the book
SMT, components
The translation model $P(f|e)$

From Word-based to Phrase-based models

f: En David llegueix el llibre nou.
e: David reads the new book. ✓

f: En David llegueix 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 llegheix el llibre nou.
e: David reads the new book. ✓

f: En David llegheix el llibre de nou.
e: David reads the book of new. X
e: David reads the book 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 caught.
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.
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.
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 $P(f|e)$

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.
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.
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:

<table>
<thead>
<tr>
<th>When are you coming back home?</th>
<th>Quan tornes</th>
<th>tu</th>
<th>a</th>
<th>casa</th>
<th>?</th>
</tr>
</thead>
</table>

(Quan tornes, When are you coming back)
SMT, components
The translation model $P(f|e)$

Phrase extraction through an example:

| When | are | you | coming | back | home | Quan tornes | tu | a | casa | ? |

(Quan tornes, When are you coming back)
Phrases extraction through an example:

(Quan tornes, When are you coming back)
(Quan tornes tu, When are you coming back)
### Intersection

<table>
<thead>
<tr>
<th>When</th>
<th>are you coming back home</th>
<th>casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quan tornes a casa</td>
<td>?</td>
<td>10 phrases</td>
<td></td>
</tr>
</tbody>
</table>
**SMT, components**

The translation model $P(f|e)$

**Intersection**

<table>
<thead>
<tr>
<th></th>
<th>Quan</th>
<th>tornes</th>
<th>a</th>
<th>casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>When</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>are</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>you</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>coming</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>back</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>home</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>?</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(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**

<table>
<thead>
<tr>
<th></th>
<th>Quan</th>
<th>tornes</th>
<th>a</th>
<th>casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>When</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>are</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>you</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>coming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>back</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>home</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(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 $P(\text{f}|\text{e})$

#### Intersection

<table>
<thead>
<tr>
<th></th>
<th>Quan</th>
<th>tornes</th>
<th>a</th>
<th>casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>When</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>are</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>you</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>coming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>back</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>home</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(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
### Intersection

<table>
<thead>
<tr>
<th></th>
<th>Quan</th>
<th>tornes</th>
<th>a</th>
<th>casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>When</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>are</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>you</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>coming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>back</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>home</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(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**

<table>
<thead>
<tr>
<th></th>
<th>Quan</th>
<th>tornes</th>
<th>a</th>
<th>casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>When</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>are</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>you</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>coming</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>back</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>home</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>?</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(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

<table>
<thead>
<tr>
<th></th>
<th>Quan</th>
<th>tornes</th>
<th>a casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>When</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>are</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>you</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>coming</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>back</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>home</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(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

<table>
<thead>
<tr>
<th></th>
<th>Quan</th>
<th>tornes</th>
<th>a</th>
<th>casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>When</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>are</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>you</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>coming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>back</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>home</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(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**

<table>
<thead>
<tr>
<th></th>
<th>Quan</th>
<th>tornes</th>
<th>a</th>
<th>casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>When</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>are</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>you</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>coming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>back</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>home</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(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

<table>
<thead>
<tr>
<th></th>
<th>Quan</th>
<th>tornes</th>
<th>a casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>When</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>are</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>you</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>coming</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>back</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>home</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>?</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(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

<table>
<thead>
<tr>
<th>When</th>
<th>Quan</th>
<th>tornes</th>
<th>a</th>
<th>casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>are</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>you</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>coming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>back</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>home</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(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 $P(f|e)$*

<table>
<thead>
<tr>
<th>Union</th>
</tr>
</thead>
<tbody>
<tr>
<td>When</td>
</tr>
<tr>
<td>are</td>
</tr>
<tr>
<td>you</td>
</tr>
<tr>
<td>coming</td>
</tr>
<tr>
<td>back</td>
</tr>
<tr>
<td>home</td>
</tr>
</tbody>
</table>

(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

<table>
<thead>
<tr>
<th></th>
<th>Quan</th>
<th>tornes</th>
<th>a</th>
<th>casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>When</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>are</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>you</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>coming</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>back</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>home</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>?</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(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

<table>
<thead>
<tr>
<th>When are you coming back home?</th>
<th>Quan tornes a casa?</th>
<th>21 phrases</th>
</tr>
</thead>
<tbody>
<tr>
<td>When</td>
<td>Quan</td>
<td></td>
</tr>
<tr>
<td>are</td>
<td>tornes</td>
<td></td>
</tr>
<tr>
<td>you</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>coming</td>
<td>casa</td>
<td></td>
</tr>
<tr>
<td>back</td>
<td>?</td>
<td></td>
</tr>
</tbody>
</table>

(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 $P(f|e)$

<table>
<thead>
<tr>
<th>Union</th>
<th>Quan</th>
<th>tornes</th>
<th>a casa</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>When are you coming back home ?</td>
<td>(Quan, When)</td>
<td>(Quan tornes, When are)</td>
<td>(Quan tornes, When are you coming)</td>
<td>(Quan tornes, When are you coming back)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

21 phrases
Phrase extraction

- The number of extracted phrases depends on the symmetrisation method.
  - Intersection: few precise phrases.
  - Union: lots of (less?) precise phrases.

- Usually, neither intersection nor union are used, but something in between.
  - Start from the intersection and add points belonging to the union according to heuristics.
Phrase extraction

- For each phrase-pair \((f_i, e_i)\), \(P(f_i|e_i)\) is estimated by frequency counts in the parallel corpus.

- The set of possible phrase-pairs conforms the set of translation options.

- The set of phrase-pairs together with their probabilities conform the translation table.
SMT, components
The translation model $P(f|e)$

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

$$T(f) = \hat{e} = \arg\max_e P(e) P(f|e)$$

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.**
SMT, components

Decoder

\[ T(f) = \hat{e} = \arg\max_e P(e) P(f|e) \]

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.**
SMT, components

Decoder

\[
T(f) = \hat{e} = \arg\max_e P(e) P(f|e)
\]

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..
Core algorithm

1. Collect translation options
2. Initial state: empty hypothesis
3. Expand hypotheses with all translation options
4. Estimate the cost for each hypothesis
5. If all source words are covered, return translation: hypothesis with the lowest cost
   - NO: go back to step 3
   - YES: return translation: hypothesis with the lowest cost
Example: Quan tornes a casa

Translation options:

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

- Translation options:
  (Quan, When)
  (Quan tornes, When are you coming back)
  (Quan tornes a casa, When are you coming back home)
  (tornes, come back)
  (tornes a casa, come back home)
  (a casa, home)

- Notation for hypotheses in construction:
  Constructed sentence so far: come back
  Source words already translated: - x - -
Example: Quan tornes a casa

Translation options:

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

Notation for hypotheses in construction:

Constructed sentence so far: come back
Source words already translated: - x - -
Example: Quan tornes a casa

- Translation options:
  - (Quan, When)
  - (Quan tornes, When are you coming back)
  - (Quan tornes a casa, When are you coming back home)
  - (tornes, come back)
  - (tornes a casa, come back home)
  - (a casa, home)

- Initial hypothesis
  
  Constructed sentence so far: $\phi$
  
  Source words already translated: - - - -
SMT, components

A beam-search decoder

φ

- - - -
When are you coming back home?

home | -- -- --

come back | - x - -

come back home | - x x x

When are you coming back | -- -- --

When | x -- --
SMT, components
A beam-search decoder

\[ \phi \]

\[ \text{When}\]
\[ x - - - \]

\[ \text{When\_are\_you\_coming\_back}\]
\[ x x - - \]

\[ \text{When\_are\_you\_coming\_back\_home}\]
\[ * x x x * \]

\[ \text{come\_back}\]
\[ - x - - \]

\[ \text{come\_back\_home}\]
\[ - x x x \]

\[ \text{home}\]
\[ -- x x \]
When are you coming back home

A beam-search decoder

SMT, components
SMT, components
A beam-search decoder

When you are coming home

When do you come back home

When are you coming back home

When are you coming back?

When you come back home
SMT, components
A beam-search decoder

When|come_back|home
   *x x x x*

When|come_back|when
   - x x x
   *x x x x*

When|come_back|home
   x x - -
   *x x x x*

When_are_you_coming_back|come_back|home
   x x - -
   *x x x x*

When_are_you_coming_back|home
   *x x x x*

come_back|home
   - x x x
   *x x x x*

come_back|when
   x x - -
   *x x x x*

come_back_home
   *x x x x*

home|come_back|when
   - x x x
   *x x x x*

home|when
   x - x x
   *x x x x*
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.

- **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

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

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

- Risk-free operation. The lowest cost translation is still there.
- But the space of hypothesis is not reduced enough.
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
6. MT Evaluation
SMT, the log-linear model

Motivation

Maximum likelihood (ML)

\[ \hat{e} = \arg\max_e P(e|f) = \arg\max_e P(e) P(f|e) \]

Maximum entropy (ME)

\[ \hat{e} = \arg\max_e P(e|f) = \arg\max_e \exp \left\{ \sum \lambda_m h_m(f, e) \right\} \]

Log-linear model

\[ \hat{e} = \arg\max_e \log P(e|f) = \arg\max_e \sum \lambda_m h_m(f, e) \]
SMT, the log-linear model

Motivation

**Maximum likelihood (ML)**

\[ \hat{e} = \arg\max_e P(e | f) = \arg\max_e P(e) P(f | e) \]

**Maximum entropy (ME)**

\[ \hat{e} = \arg\max_e P(e | f) = \arg\max_e \exp \left\{ \sum \lambda_m h_m(f, e) \right\} \]

**Log-linear model**

\[ \hat{e} = \arg\max_e \log P(e | f) = \arg\max_e \sum \lambda_m h_m(f, e) \]
SMT, the log-linear model

Motivation

Maximum likelihood (ML)

\[ \hat{e} = \arg\max_e P(e|f) = \arg\max_e P(e) P(f|e) \]

Maximum entropy (ME)

\[ \hat{e} = \arg\max_e P(e|f) = \arg\max_e \exp \left\{ \sum \lambda_m h_m(f, e) \right\} \]

\[ \hat{e} = \arg\max_e \log P(e|f) = \arg\max_e \sum \lambda_m h_m(f, e) \]

Log-linear model
SMT, the log-linear model

Motivation

Maximum likelihood (ML)

\[ \hat{e} = \arg \max_e P(e|f) = \arg \max_e P(e) P(f|e) \]

Maximum entropy (ME)

\[ \hat{e} = \arg \max_e \log P(e|f) = \arg \max_e \sum \lambda_m h_m(f, e) \]

Log-linear model with

\[ h_1(f, e) = \log P(e), \quad h_2(f, e) = \log P(f|e), \quad \text{and} \quad \lambda_1 = \lambda_2 = 1 \]

⇒ 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 $\lambda_m$ 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: $P(e)$, $P(f|e)$, $P(e|f)$, $\text{lex}(f|e)$, $\text{lex}(e|f)$, $\text{ph}(e)$, $w(e)$ and $P_d(e, f)$.

- Language model $P(e)$
  $P(e)$: Language model probability as in ML model.

- Translation model $P(f|e)$
  $P(f|e)$: Translation model probability as in ML model.

- Translation model $P(e|f)$
  $P(e|f)$: Inverse translation model probability to be added to the generative one.
SMT, the log-linear model

Features

State of the art feature functions

Eight features are usually used: \( P(e), P(f|e), P(e|f), \, \text{lex}(f|e), \, \text{lex}(e|f), \, \text{ph}(e), \, w(e) \) and \( P_d(e, f) \).

- Translation model \( \text{lex}(f|e) \)
  \( \text{lex}(f|e) \): Lexical translation model probability.

- Translation model \( \text{lex}(e|f) \)
  \( \text{lex}(e|f) \): Inverse lexical translation model probability.

- Phrase penalty \( \text{ph}(e) \)
  \( \text{ph}(e) \): A constant cost per produced phrase.
SMT, the log-linear model
Features

State of the art feature functions

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

- **Word penalty** $w(e)$
  $w(e)$: A constant cost per produced word.

- **Distortion** $P_d(e, f)$
  $P_d(\text{ini}_{phrase_i}, \text{end}_{phrase_{i-1}})$: Relative distortion probability distribution. A simple distortion model:
  $$P_d(\text{ini}_{phrase_i}, \text{end}_{phrase_{i-1}}) = \alpha|\text{ini}_{phrase_i} - \text{end}_{phrase_{i-1}} - 1|$$
Development training, weights optimisation

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

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

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

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

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

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

1. \( \lambda_i \) initialisation
2. Select a direction \( k \), fix \( \lambda_i \neq \lambda_k \)
3. Best \( \lambda_k \) (line minimisation)
4. all \( k \) explored?
5. \( \lambda_i \) stable?
6. Optimal \( \lambda_i \)
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
6. MT Evaluation
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.

\[(\text{word}) \Rightarrow (\text{word}, \text{lemma}, \text{PoS}, \text{morphology}, ...)\]

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.

\[(\text{word}) \rightarrow (\text{word}, \text{lemma}, \text{PoS}, \text{morphology}, \ldots)\]

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

\[
\begin{align*}
\text{lemma}_f & \downarrow T & \text{PoS}_f & \downarrow T & \text{morphology}_f & \downarrow T & \text{word}_f \\
\text{lemma}_e & \downarrow T & \text{PoS}_e & \downarrow T & \text{morphology}_e & & \text{word}_e
\end{align*}
\]
Factored translation models

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

\[(\text{word}) \rightarrow (\text{word, lemma, PoS, morphology, ...})\]

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

- \(casa_f\) \(\downarrow T\) \(house_e\)
- \(NN_f\) \(\downarrow T\) \(NN_e\)
- \(fem.,\ plural_f\) \(\downarrow T\) \(plural_e\)
- \(cases_f\) \(\rightarrow G\) \(houses_e\)
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.
SMT, beyond standard SMT
Syntactic translation models

**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.
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

```
S
  | V
  | llegeix
NP
  | PP
  | NN
  | En David

S
  | V
  | VP
  | DT
  | NN
  | PP
  | el llibre nou
```

```
S
  | V
  | VP
  | NP
  | NN
  | DT
  | PP
  | NN
```
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
Ongoing 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.
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 syntax 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.
Outline

1. Introduction
2. Basics
3. Components
4. The log-linear model
5. Beyond standard SMT
6. MT Evaluation
MT Evaluation
Importance for system development

Implementation

Error detection
Error analysis
Refinement

Evaluation methods

Test

YES OK? NO

Unfruitful results
MT Evaluation
Importance for system development

- Error detection
- Error analysis
- Refinement
- Implementation
- Test
- Evaluation methods
- Unfruitful results

Flowchart:
- Implementation → Test
- Test: YES → OK
- Test: NO → Unfruitful results
- Error detection → Error analysis → Refinement → Implementation

Diagram:
- Evaluation methods
- Unfruitful results
- YES → OK
- NO
- Error detection
- Error analysis
- Refinement
- Implementation
- Test

Flowchart:
- YES → OK
- NO → Unfruitful results
- Error detection
- Error analysis
- Refinement
- Implementation
- Test

Diagram:
- Evaluation methods
- Unfruitful results
- YES → OK
- NO
- Error detection
- Error analysis
- Refinement
- Implementation
- Test
MT Evaluation
Importance for system development
MT Evaluation

Importance for system development

- Error detection
- Error analysis
- Refinement
- Implementation
- Test

Evaluation methods

Unfruitful results

YES OK? NO
MT Evaluation
Importance for system development

Evaluation methods

Error detection
Error analysis
Refinement
Implementation
Test
OK? YES NO Unfruitful results
MT Evaluation
Importance for system development

Error detection → Error analysis → Refinement → Implementation → Test

OK? YES → Evaluation methods
NO → Unfruitful results
MT Evaluation
Importance for system development

- Error detection
- Error analysis
- Refinement
- Implementation
- Test

Evaluation methods

OK? YES NO

Unfruitful results
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

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

\[ e: \text{ This sentence is going to be difficult to evaluate. } \]

Ref1: The evaluation of the translation 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.
Evaluation is important in the system development cycle. Automatic evaluation accelerates significantly 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.
Thanks to Jesús Giménez for some of the material
Part II

SMT experiments
Outline Part II

7 Translation system
   - Software
   - Steps

8 Evaluation system
   - Software
   - Steps
Build your own SMT system

1. Language model with SRILM.

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

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)


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)

```
```
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

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

```
cristina@cosmos:~$ train-factored-phrase-model.model.perl -help
Train Phrase Model
Steps: (--first-step to --last-step)
(1) prepare corpus
(2) run GIZA
(3) align words
(4) learn lexical translation
(5) extract phrases
(6) score phrases
(7) learn reordering model
(8) learn generation model
(9) create decoder config file
```
3. **Train the translation model** (cont’d)

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

```
bin/moses-scripts/training/train-factored-phrase-model.perl 
```

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

Steps

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
   ```
5. **Run Moses decoder on a test set**

1. Tokenise and lowercase 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 -config ./filteredmodel/moses.ini
   -input-file testset.es > testset.translated.en
   ```
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 IQmt package.
   http://www.lsi.upc.edu/~nlp/IQMT/
1. Evaluate the results

With BLEU scoring tool in Moses:

moses/scripts/generic/multi-bleu.perl references.en < testset.translated.en
2. Evaluate the results on-line

OpenMT Evaluation Demo

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

Appendix: 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]
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]
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\textsuperscript{+}09]
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
- Koehn, 2006
- Way & Hassan, 2009
- Lopez, 2008 [Lop08]
- Giménez, 2009 [Gim08]


References

David Chiang.
A hierarchical phrase-based model for statistical machine translation.

David Chiang, Kevin Knight, and Wei Wang.
11,001 new features for statistical machine translation.

Josep Mª Crego and José B. Mari no.
Improving smt by coupling reordering and decoding.

Sander Canisius and Antal van den Bosch.
A constraint satisfaction approach to machine translation.


Andrew Finch and Eiichiro Sumita.  
Bidirectional phrase-based statistical machine translation.  

Jesús Giménez and Enrique Amigó.  

Jesús Giménez.  

Jesús Giménez and Lluís Màrquez.  
*Discriminative Phrase Selection for SMT*, pages 205–236.  

Hany Hassan, Khalil Sima’an, and Andy Way.  
A syntactified direct translation model with linear-time decoding.  


Franz Josef Och and Hermann Ney.  
The alignment template approach to statistical machine translation.  

Franz Josef Och, Christoph Tillmann, and Hermann Ney.  
Improved alignment models for statistical machine translation.  

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu.  

Jonathan Slocum.  
