Grammatical Inference: News from the Machine Translation Front

François Yvon


LIMSI-CNRS & Université Paris-Sud 11

September 22, 2008
SMT = corpora + machine learning algorithms
SMT
  =
  large corpora
  +
  simple machine learning algorithms
Outline

SMT for restricted domain and look-alike languages

= large corpora

+ simple machine learning algorithms
Outline

General SMT
    =
    linguistically analyzed corpora
    +
    structure aware machine learning algorithms
Some problems with machine translation
Is machine translation possible at all?

\[
f = \text{Ich werde Ihnen die entsprechenden Anmerkungen aushändigen}
\]

\[
e = \text{I will pass on to you the corresponding comments}
\]
Mainstream Statistical Machine Translation

Introducing Phrase-Based Statistical Machine Translation
Mainstream Statistical Machine Translation

Introducing Phrase-Based Statistical Machine Translation

1. take a set of parallel sentences (*bitext*)
   ▶ align each pair \((f, e)\), word for word
   ▶ train translation model: the “phrase” table \(\{(f, e)\}\)

2. take a set of monolingual texts
   ▶ train statistical target language model

3. make sure to tune your system

4. translate \(f = \text{solve}\)

\[
\arg\max_{e \in E} s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f)
\]

5. and get some numbers

6. not happy ? goto 1
Mainstream Statistical Machine Translation
Introducing Phrase-Based Statistical Machine Translation

1. take a set of parallel sentences (*bitext*)
   - align each pair \((f,e)\), word for word
   - train translation model: the “phrase” table \([((f, e))\]

2. take a set of monolingual texts
   - train statistical target language model

3. make sure to tune your system

4. translate \(f = \text{solve}\)

\[
\arg\max_{e \in E} s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f)
\]

5. and get some numbers

6. not happy? goto 1
Mainstream Statistical Machine Translation

Introducing Phrase-Based Statistical Machine Translation

1. take a set of parallel sentences (*bitext*)
   - align each pair (f,e), word for word
   - train translation model: the “phrase” table \{(f, e)\}

2. take a set of monolingual texts
   - train statistical target language model

3. make sure to tune your system

4. translate \( f = \text{solve} \)

\[
\arg\max_{e \in E} s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f)
\]

5. and get some numbers

6. not happy ? goto 1
Mainstream Statistical Machine Translation

Introducing Phrase-Based Statistical Machine Translation

1. take a set of parallel sentences (bitext)
   - align each pair (f,e), word for word
   - train translation model: the “phrase” table \{ (f, e) \}

2. take a set of monolingual texts
   - train statistical target language model

3. make sure to tune your system

4. translate \( f = \text{solve} \)

\[
\arg\max_{e \in E} s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f)
\]

5. and get some numbers

6. not happy? goto 1
Mainstream Statistical Machine Translation

Introducing Phrase-Based Statistical Machine Translation

1. take a set of parallel sentences (*bitext*)
   - align each pair (*f*, *e*), word for word
   - train translation model: the “phrase” table \{
     (f, e)\}

2. take a set of monolingual texts
   - train statistical target language model

3. make sure to tune your system

4. translate \( f = \text{solve} \)

\[
\arg\max_{e \in E} s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f)
\]

5. and get some numbers

6. not happy? goto 1
Mainstream Statistical Machine Translation
Introducing Phrase-Based Statistical Machine Translation

1. take a set of parallel sentences (*bitext*)
   ▶ align each pair \((f, e)\), word for word
   ▶ train translation model: the “phrase” table \(\{(f, e)\}\)

2. take a set of monolingual texts
   ▶ train statistical target language model

3. make sure to tune your system

4. translate \(f = \text{solve}\)

\[
\arg\max_{e \in E} s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f)
\]

5. and get some numbers

6. not happy? goto 1
Mainstream Statistical Machine Translation
Introducing Phrase-Based Statistical Machine Translation

1. take a set of parallel sentences (*bitext*)
   - align each pair (f,e), word for word
   - train translation model: the “phrase” table {((f, e))}

2. take a set of monolingual texts
   - train statistical target language model

3. make sure to tune your system

4. translate \( f = \text{solve} \)

\[
\text{argmax}_{e \in E} s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f)
\]

5. and get some numbers

6. not happy? goto 1
Mainstream Statistical Machine Translation
Introducing Phrase-Based Statistical Machine Translation

1. take a set of parallel sentences (bitext)
   ▶ align each pair \((f, e)\), word for word
   ▶ train translation model: the “phrase” table \(\{(f, e)\}\)

2. take a set of monolingual texts
   ▶ train statistical target language model

3. make sure to tune your system

4. translate \(f = \text{solve}\)

\[
\text{argmax } s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f)
\]

5. and get some numbers

6. not happy ? goto 1
Take a set of parallel sentences

- bilingual corpus, per sentence alignment

\[ f = \textit{Pourquoi donc les producteurs d’armes de l’UE devraient-ils s’enrichir sur le dos de personnes innocentes ?} \]
\[ e = \textit{So why should EU arms producers profit at the expense of innocent people ?} \]

- Main sources:
  - documents from multilingual institutions, literature, touristic guides, technical documentations
  - news, web sites, blogs, speech transcripts

- Not enough? Mine comparable corpora (eg. [26])

Large corpora available, yet data scarcity still a serious bottleneck
Take a set of parallel sentences

- bilingual corpus, per sentence alignment
- Main sources:
  - documents from multilingual institutions, literature, touristic guides, technical documentations
  - news, web sites, blogs, speech transcripts
- Not enough? Mine comparable corpora (eg. [26])

Large corpora available, yet data scarcity still a serious bottleneck
Take a set of parallel sentences

- bilingual corpus, per sentence alignment
- Main sources:
  - documents from multilingual institutions, literature, touristic guides, technical documentations
  - news, web sites, blogs, speech transcripts
- Not enough? Mine comparable corpora (eg. [26])

Large corpora available, yet data scarcity still a serious bottleneck
Take a set of parallel sentences

- bilingual corpus, per sentence alignment
- Main sources:
  - documents from multilingual institutions, literature, touristic guides, technical documentations
  - news, web sites, blogs, speech transcripts
- Not enough? Mine comparable corpora (eg. [26])

Large corpora available, yet data scarcity still a serious bottleneck
Take a set of parallel sentences

- bilingual corpus, per sentence alignment
- Main sources:
  - documents from multilingual institutions, literature, touristic guides, technical documentations
  - news, web sites, blogs, speech transcripts
- Not enough? Mine comparable corpora (eg. [26])

Large corpora available, yet data scarcity still a serious bottleneck
Take a set of parallel sentences

- bilingual corpus, per sentence alignment
- Main sources:
  - documents from multilingual institutions, literature, touristic guides, technical documentations
  - news, web sites, blogs, speech transcripts
- Not enough? Mine comparable corpora (eg. [26])

Large corpora available, yet data scarcity still a serious bottleneck
Take a set of parallel sentences

- bilingual corpus, per sentence alignment
- Main sources:
  - documents from multilingual institutions, literature, touristic guides, technical documentations
  - news, web sites, blogs, speech transcripts
- Not enough? Mine comparable corpora (eg. [26])

Large corpora available, yet data scarcity still a serious bottleneck
null
i believe that the principle of relative stability is a fundamental legal principle of the common fisheries policy.

à mon sens, le principe de stabilité relative est un principe juridique fondamental de la politique commune de la pêche.
I believe that the principle of relative stability is a fundamental legal principle of the common fisheries policy.
null
i
believe
that
the
principle
of
relative
stability
is
a
fundamental
legal
principle
of
the
common
fisheries
policy
à
mon
sens,
le
principe
de
stabilité
relative
est
un
principe
juridique
fondamental
de
la
politique
commune
de
la
pêche
I believe that the principle of relative stability is a fundamental legal principle of the common fisheries policy.
Training 1.a: build word alignments
A more noisy case

NULL
it
seems
absolutely
disgraceful
that
we
pass
legislation
and
do
not
adhere
to
it
ourselves
.

nous
votons
des
réglementations
et
nous
ne
nous
y
conformons
même
pas
‘
c
‘est
scandaleux
.

it seems absolutely disgraceful that we pass legislation and do not adhere to it ourselves.
A more noisy case

It seems absolutely disgraceful that we pass legislation and do not adhere to it ourselves, c'est scandaleux.

Nous votons des réglementations et nous ne nous conformons même pas, c'est scandaleux.
Training 1.a: build word alignments
A more noisy case

it seems absolutely disgraceful that we pass legislation and do not adhere to it ourselves, it's scandalous.

NULL
it seems absolutely disgraceful that we pass legislation and do not adhere to it ourselves

it seems absolutely disgraceful that we pass legislation and do not adhere to it ourselves

it seems absolutely disgraceful that we pass legislation and do not adhere to it ourselves
Training 1.a: build word alignments
A more noisy case

It seems absolutely disgraceful that we pass legislation and do not adhere to it ourselves.

It seems scandalous that we pass legislation and do not adhere to it ourselves.
Training 1.a: build word alignments

- asymmetric (= many-to-one) alignments (IBM1-IBM5 [6], HMMs [36])
  - train: estimate $P(a, f|e)$ (EM like)
  - align: $a^* = \arg\max P(a|f, e) = \arg\max P(a, f|e)$
  - translate:
    $e^* = \arg\max_e P(f|e)P(e) = \arg\max_e P(e)\arg\max_a P(a, f|e)$

- public domain implementations (Giza++ [28] ; MTTK [13])
- discriminative training (and many more features) helps a bit [24, 1, 4]
- but supervision data is scarce and unreliable
  for asymmetric models, an almost solved issue?
Training 1.a: build word alignments

► asymmetric (= many-to-one) alignments (IBM1-IBM5 [6], HMMs [36])
  ► train: estimate $P(a, f|e)$ (EM like)
  ► align: $a^* = \text{argmax } P(a|f, e) = \text{argmax } P(a, f|e)$
  ► translate:
    $e^* = \text{argmax}_e P(f|e)P(e) = \text{argmax}_e P(e)\text{argmax}_a P(a, f|e)$
  ► public domain implementations (Giza++ [28] ; MTTK [13])
  ► discriminative training (and many more features) helps a bit [24, 1, 4]
  ► but supervision data is scarce and unreliable

for asymmetric models, an almost solved issue ?
Training 1.a: build word alignments

- asymmetric (= many-to-one) alignments (IBM1-IBM5 [6], HMMs [36])
  - train: estimate $P(a, f|e)$ (EM like)
  - align: $a^* = \arg\max P(a|f, e) = \arg\max P(a, f|e)$
  - translate:
    $e^* = \arg\max_e P(f|e)P(e) = \arg\max_e P(e)\arg\max_a P(a, f|e)$

- public domain implementations (Giza++ [28] ; MTTK [13])
- discriminative training (and many more features) helps a bit [24, 1, 4]
- but supervision data is scarce and unreliable

for asymmetric models, an almost solved issue?
Training 1.a: build word alignments

- asymmetric (= many-to-one) alignments (IBM1-IBM5 [6], HMMs [36])
  - train: estimate $P(a, f|e)$ (EM like)
  - align: $a^* = \arg\max P(a|f, e) = \arg\max P(a, f|e)$
  - translate:
    $e^* = \arg\max_e P(f|e)P(e) = \arg\max_e P(e) \arg\max_a P(a, f|e)$

- public domain implementations (Giza++ [28]; MTTK [13])
- discriminative training (and many more features) helps a bit [24, 1, 4]
- but supervision data is scarce and unreliable for asymmetric models, an almost solved issue?
Training 1.a: build word alignments

- asymmetric (= many-to-one) alignments (IBM1-IBM5 [6], HMMs [36])
  - train: estimate $P(a, f|e)$ (EM like)
  - align: $a^* = \arg\max P(a|f, e) = \arg\max P(a, f|e)$
  - translate:
    $$e^* = \arg\max_e P(f|e)P(e) = \arg\max_e P(e)\arg\max_a P(a, f|e)$$

- public domain implementations (Giza++ [28]; MTTK [13])
- discriminative training (and many more features) helps a bit [24, 1, 4]
- but supervision data is scarce and unreliable

for asymmetric models, an almost solved issue?
Training 1.a: build word alignments

- asymmetric (= many-to-one) alignments (IBM1-IBM5 [6], HMMs [36])
  - train: estimate $P(a, f|e)$ (EM like)
  - align: $a^* = \arg\max P(a|f, e) = \arg\max P(a, f|e)$
  - translate:
    $$e^* = \arg\max_e P(f|e)P(e) = \arg\max_e P(e) \arg\max_a P(a, f|e)$$

- public domain implementations (Giza++ [28]; MTTK [13])
- discriminative training (and many more features) helps a bit [24, 1, 4]
- but supervision data is scarce and unreliable for asymmetric models, an almost solved issue?
Training 1.a: build word alignments

- asymmetric (= many-to-one) alignments (IBM1-IBM5 [6], HMMs [36])
  - train: estimate $P(a, f|e)$ (EM like)
  - align: $a^* = \arg\max P(a|f, e) = \arg\max P(a, f|e)$
  - translate:
    $e^* = \arg\max_e P(f|e)P(e) = \arg\max_e P(e)\arg\max_a P(a, f|e)$

- public domain implementations (Giza++ [28] ; MTTK [13])
- discriminative training (and many more features) helps a bit [24, 1, 4]
- but supervision data is scarce and unreliable

For asymmetric models, an almost solved issue?
Training 1.a: build word alignments

- asymmetric (= many-to-one) alignments (IBM1-IBM5 [6], HMMs [36])
  - train: estimate $P(a, f|e)$ (EM like)
  - align: $a^* = \arg\max P(a|f, e) = \arg\max P(a, f|e)$
  - translate:
    $e^* = \arg\max_e P(f|e)P(e) = \arg\max_e P(e)\arg\max_a P(a, f|e)$

- public domain implementations (Giza++ [28] ; MTTK [13])
- discriminative training (and many more features) helps a bit [24, 1, 4]
- but supervision data is scarce and unreliable

for asymmetric models, an almost solved issue?
Training 1.a: build word alignments

- asymmetric (= many-to-one) alignments (IBM1-IBM5 [6], HMMs [36])
  - train: estimate $P(a, f|e)$ (EM like)
  - align: $a^* = \arg\max P(a|f, e) = \arg\max P(a, f|e)$
  - translate:
    $$e^* = \arg\max_e P(f|e)P(e) = \arg\max_e P(e)\arg\max_a P(a, f|e)$$

- public domain implementations (Giza++ [28] ; MTTK [13])
- discriminative training (and many more features) helps a bit [24, 1, 4]
- but supervision data is scarce and unreliable

for asymmetric models, an almost solved issue ?
Training 1.b: accumulate “phrases” and their statistics

\[ f = \text{michael geht davon aus, dass er im hause bleibt} \]

\[ e = \text{michael assumes he that will stay in the hause} \]

(example from P. Koehn)

A symmetrized alignment
Training 1.b: accumulate “phrases” and their statistics

\( f = \text{michael geht davon aus, dass er im hause bleibt} \)

\( e = \text{michael assumes he that will stay in the hause} \)

(example from P. Koehn)
Training 1.b: accumulate “phrases” and their statistics

\[ f = \text{michael geht davon aus, dass er im hause bleibt} \]
\[ e = \text{michael assumes he that will stay in the hause} \]
(Example from P. Koehn)

\[ N(\text{michael assumes} ; \text{michael geht davon aus}) \]
Training 1.b: accumulate “phrases” and their statistics

\( f = \text{michael geht davon aus, dass er im hause bleibt} \)

\( e = \text{michael assumes he that will stay in the hause} \)

(example from P. Koehn)

\[
\begin{array}{cccccc}
\text{michael} & \text{geht} & \text{davon} & \text{aus} & \text{er} & \text{im} & \text{hause} & \text{bleibt} \\
\text{house} & \text{the} & \text{in} & \text{stay} & \text{will} & \text{he} & \text{that} & \text{assumes} \\
\text{in} & \text{stay} & \text{will} & \text{he} & \text{that} & \text{assumes} & \text{michael} \\
\end{array}
\]

\[
N(\text{michael assumes that ; michael geht davon aus , dass}) + +
\]
Training 1.b: accumulate “phrases” and their statistics

\[ f = \text{michael geht davon aus, dass er im hause bleibt} \]
\[ e = \text{michael assumes he that will stay in the hause} \]

(example from P. Koehn)

\[ N(he \ will \ stay; \ er \ him \ hause \ bleibt) += 0 \]
Training 1.b: accumulate “phrases” and their statistics

\[ f = \text{michael geht davon aus, dass er im hause bleibt} \]

\[ e = \text{michael assumes he that will stay in the hause} \]

(example from P. Koehn)

\[
\begin{array}{cccccc}
\text{michael} & \text{geht} & \text{davon} & \text{aus} & \text{dass} & \text{er} & \text{im} & \text{hausa} & \text{bleibt} \\
\text{house} & \text{the} & \text{in} & \text{stay} & \text{will} & \text{he} & \text{that} & \text{assumes} & \text{michael} \\
\end{array}
\]

\[ N(\text{stay in the house} ; \text{im hause bleibt})++ \]
Training 1.b : accumulate “phrases” and their statistics

- translation model = “phrase” table \{(e, f), w(e, f) = P(f|e)\}
- crudely heuristic and very noisy
  - forced alignment of non aligned words
  - non literal translations
- sparsity: smoothing \( P(f|e) = \frac{N(e,f)}{N(e)} \) helps [40, 15]
- linguistics does not help [20]
- size an issue ? pruning helps runtimes [16]
- size NOT an issue ? Use gappy phrases [9]

The largest the phrase table, the better the translation
Training 1.b: accumulate “phrases” and their statistics

- translation model = “phrase” table \( \{(e, f), w(e, f) = P(f|e)\} \)
- crudely heuristic and very noisy
  - forced alignment of non aligned words
  - non literal translations
- sparsity: smoothing \( P(f|e) = \frac{N(e,f)}{N(e)} \) helps [40, 15]
- linguistics does not help [20]
- size an issue? pruning helps runtimes [16]
- size NOT an issue? Use gappy phrases [9]

The largest the phrase table, the better the translation
Training 1.b: accumulate “phrases” and their statistics

- translation model = “phrase” table \( \{(e, f), w(e, f) = P(f|e)\} \)

- crudely heuristic and very noisy
  - forced alignment of non aligned words
  - non literal translations

- sparsity: smoothing \( P(f|e) = \frac{N(e,f)}{N(e)} \) helps [40, 15]

- linguistics does not help [20]

- size an issue? pruning helps runtimes [16]

- size NOT an issue? Use gappy phrases [9]

The largest the phrase table, the better the translation
Training 1.b: accumulate “phrases” and their statistics

- Translation model = “phrase” table \{{{e, f}, w(e, f) = P(f|e)}\}
- Crudely heuristic and very noisy
  - Forced alignment of non aligned words
  - Non literal translations
- Sparsity: smoothing \( P(f|e) = \frac{N(e,f)}{N(e)} \) helps [40, 15]
- Linguistics does not help [20]
- Size an issue? Pruning helps runtimes [16]
- Size NOT an issue? Use gappy phrases [9]

The largest the phrase table, the better the translation
Training 1.b: accumulate “phrases” and their statistics

- Translation model = “phrase” table \( \{(e, f), w(e, f) = P(f|e)\} \)
- Crudely heuristic and very noisy
  - Forced alignment of non-aligned words
  - Non literal translations
- Sparsity: smoothing \( P(f|e) = \frac{N(e,f)}{N(e)} \) helps [40, 15]
- Linguistics does not help [20]
- Size an issue? Pruning helps runtimes [16]
- Size NOT an issue? Use gappy phrases [9]

The largest the phrase table, the better the translation
Training 1.b: accumulate “phrases” and their statistics

- translation model = “phrase” table \((e, f), w(e, f) = P(f|e)\)
- crudely heuristic and very noisy
  - forced alignment of non aligned words
  - non literal translations
- sparsity: smoothing \(P(f|e) = \frac{N(e,f)}{N(e)}\) helps [40, 15]
- linguistics does not help [20]
  - size an issue? pruning helps runtimes [16]
  - size NOT an issue? Use gappy phrases [9]

The largest the phrase table, the better the translation
Training 1.b: accumulate “phrases” and their statistics

- translation model = “phrase” table \( \{(e, f), w(e, f) = P(f|e)\} \)

- crudely heuristic and very noisy
  - forced alignment of non aligned words
  - non literal translations

- sparsity: smoothing \( P(f|e) = \frac{N(e,f)}{N(e)} \) helps [40, 15]

- linguistics does not help [20]

- size an issue? pruning helps runtimes [16]

- size NOT an issue? Use gappy phrases [9]

The largest the phrase table, the better the translation
Training 1.b: accumulate “phrases” and their statistics

- Translation model = “phrase” table \( \{(e, f), w(e, f) = P(f|e)\} \)
- Crudely heuristic and very noisy
  - Forced alignment of non aligned words
  - Non literal translations
- Sparsity: smoothing \( P(f|e) = \frac{N(e,f)}{N(e)} \) helps [40, 15]
- Linguistics does not help [20]
- Size an issue? Pruning helps runtimes [16]
- Size NOT an issue? Use gappy phrases [9]

The largest the phrase table, the better the translation.
Training 1.b: accumulate “phrases” and their statistics

- Translation model = “phrase” table \( \{(e, f), w(e, f) = P(f|e)\} \)
- Crudely heuristic and very noisy
  - Forced alignment of non aligned words
  - Non literal translations
- Sparsity: smoothing \( P(f|e) = \frac{N(e,f)}{N(e)} \) helps [40, 15]
- Linguistics does not help [20]
- Size an issue? Pruning helps runtimes [16]
- Size NOT an issue? Use gappy phrases [9]

The largest the phrase table, the better the translation
Training 1.b: accumulate “phrases” and their statistics

- translation model = “phrase” table \(((e, f), w(e, f) = P(f|e))\)
- crudely heuristic and very noisy
  - forced alignment of non aligned words
  - non literal translations
- sparsity: smoothing \(P(f|e) = \frac{N(e,f)}{N(e)}\) helps [40, 15]
- linguistics does not help [20]
- size an issue? pruning helps runtimes [16]
- size NOT an issue? Use gappy phrases [9]

The largest the phrase table, the better the translation
Training 2: learn a target language model
The same old story

- $n$-gram language models
  - large span (≥ 5-gram) models help
  - more training data helps...
  - ... much more than smart smoothing
  - ... that can’t be computed anyway

scaling up [10, 33] more important than modeling?
Training 2: learn a target language model

The same old story

- n-gram language models
- large span ($\geq$ 5-gram) models help
- more training data helps...
- ... much more than smart smoothing
- ... that can't be computed anyway

Scaling up [10, 33] more important than modeling?
Training 2: learn a target language model
The same old story

- $n$-gram language models
- large span (≥ 5-gram) models help
- more training data helps...
  - ... much more than smart smoothing
  - ... that can’t be computed anyway

Results from [5]
scaling up [10, 33] more important than modeling ?
Training 2: learn a target language model

The same old story

- $n$-gram language models
- large span ($\geq$ 5-gram) models help
- more training data helps...
- ... much more than smart smoothing
- ... that can’t be computed anyway

Results from [5]

scaling up [10, 33] more important than modeling?
Training 2: learn a target language model
The same old story

- $n$-gram language models
- large span ($\geq 5$-gram) models help
- more training data helps...
- ... much more than smart smoothing
- ... that can’t be computed anyway

scaling up [10, 33] more important than modeling?
Training 2: learn a target language model
The same old story

- $n$-gram language models
- large span ($\geq$ 5-gram) models help
- more training data helps...
- ... much more than smart smoothing
- ... that can’t be computed anyway

Results from [5]

scaling up [10, 33] more important than modeling?
Training 2: learn a target language model
The same old story

- $n$-gram language models
- large span ($\geq$ 5-gram) models help
- more training data helps...
- ... much more than smart smoothing
- ... that can’t be computed anyway

scaling up [10, 33] more important than modeling?
Training 3: tune the score function

Translation score

\[ s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f) \]

where \( F_k(e, f) \) corresponds to:

- translation models,
- language model,
- distortion models,
- length model,
- segmentation model, etc

- use held-out data \( D \) to optimize weights \( \{\lambda_k, k = 1...K\} \)

\[ \lambda^* = \arg\min_{\lambda} \text{LOSS}(D, \lambda) \] [27]

- LOSS() typically not differentiable in \( \lambda \)

doing in right makes a difference [8, 25]
Training 3: tune the score function

Translation score

\[ s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f) \]

where \( F_k(e, f) \) corresponds to:
- translation models, language model, distortion models, length model, segmentation model, etc

▶ use held-out data \( D \) to optimize weights \( \{\lambda_k, k = 1 \ldots K\} \)

\[ \lambda^* = \arg\min_{\lambda} LOSS(D, \lambda) \] [27]

▶ LOSS() typically not differentiable in \( \lambda \)

doing in right makes a difference [8, 25]
Training 3: tune the score function

Translation score

\[ s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f) \]

where \( F_k(e, f) \) corresponds to:
- translation models, language model, distortion models,
- length model, segmentation model, etc

- use held-out data \( D \) to optimize weights \( \{\lambda_k, k = 1...K\} \)

\[ \lambda^* = \arg\min_{\lambda} LOSS(D, \lambda) \] \[ [27] \]

- \( LOSS() \) typically not differentiable in \( \lambda \)

doing in right makes a difference \([8, 25]\)
Training 3: tune the score function

Translation score

\[ s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f) \]

where \( F_k(e, f) \) corresponds to:
- translation models
- language model
- distortion models
- length model
- segmentation model, etc

- use held-out data \( D \) to optimize weights \( \{\lambda_k, k = 1...K\} \)

\[ \lambda^* = \arg\min_{\lambda} \text{LOSS}(D, \lambda) [27] \]

- LOSS() typically not differentiable in \( \lambda \)

doing in right makes a difference [8, 25]
Training 3: tune the score function

Translation score

\[ s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f) \]

where \( F_k(e, f) \) corresponds to:

- translation models, language model, distortion models, length model, segmentation model, etc

- use held-out data \( D \) to optimize weights \( \{\lambda_k, k = 1 \ldots K\} \)

\[ \lambda^* = \arg\min_{\lambda} \text{LOSS}(D, \lambda) \ [27] \]

- \( \text{LOSS}() \) typically not differentiable in \( \lambda \)

doing in right makes a difference \([8, 25]\)
Decoding, an optimisation problem

Solve $\arg\max_e s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f)$

- very large hypothesis space ($\Rightarrow$ a NP-hard problem [17])
  - all segmentations of source sentence
  - all translations of each source phrase
  - every permutation of the source phrases
- heuristic search + fine-tuned pruning
- high performance, fast decoding doable

Not so much an issue ... for laboratory systems
Decoding, an optimisation problem

Solve \( \arg\max_e s(e, f) = \sum_{k=1}^K \lambda_k F_k(e, f) \)

- very large hypothesis space (⇒ a NP-hard problem [17])
  - all segmentations of source sentence
    - all translations of each source phrase
    - every permutation of the source phrases
  - heuristic search + fine-tuned pruning
  - high performance, fast decoding doable

Not so much an issue ... for laboratory systems
Decoding, an optimisation problem

\[
\text{Solve } \text{argmax}_e s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f)
\]

▶ very large hypothesis space (⇒ a NP-hard problem [17])
  ▶ all segmentations of source sentence
  ▶ all translations of each source phrase
  ▶ every permutation of the source phrases

▶ heuristic search + fine-tuned pruning
▶ high performance, fast decoding doable

Not so much an issue ... for laboratory systems
Decoding, an optimisation problem

Solve \( \arg \max_{e} s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f) \)

- very large hypothesis space (\( \Rightarrow \) a NP-hard problem [17])
  - all segmentations of source sentence
  - all translations of each source phrase
  - every permutation of the source phrases
- heuristic search + fine-tuned pruning
- high performance, fast decoding doable

Not so much an issue ... for laboratory systems
Decoding, an optimisation problem

Solve \( \arg\max_e s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f) \)

- very large hypothesis space (\( \Rightarrow \) a NP-hard problem [17])
  - all segmentations of source sentence
  - all translations of each source phrase
  - every permutation of the source phrases

- heuristic search + fine-tuned pruning

- high performance, fast decoding doable

Not so much an issue ... for laboratory systems
Decoding, an optimisation problem

Solve \( \text{argmax}_e s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f) \)

- very large hypothesis space (\( \Rightarrow \) a NP-hard problem [17])
  - all segmentations of source sentence
  - all translations of each source phrase
  - every permutation of the source phrases
- heuristic search + fine-tuned pruning
- high performance, fast decoding doable

Not so much an issue ... for laboratory systems
Decoding, an optimisation problem
Solve \( \text{argmax}_e s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f) \)

- very large hypothesis space (\( \Rightarrow \) a NP-hard problem [17])
  - all segmentations of source sentence
  - all translations of each source phrase
  - every permutation of the source phrases
- heuristic search + fine-tuned pruning
- high performance, fast decoding doable

Monotonic search

Not so much an issue ... for laboratory systems
Decoding, an optimisation problem

Solve $\arg\max_{e} s(e, f) = \sum_{k=1}^{K} \lambda_k F_k(e, f)$

- very large hypothesis space ($\Rightarrow$ a NP-hard problem [17])
  - all segmentations of source sentence
  - all translations of each source phrase
  - every permutation of the source phrases
- heuristic search + fine-tuned pruning
- high performance, fast decoding doable

Not so much an issue ... for laboratory systems
Get some numbers
Evaluating machine translation

- subjective evaluation is very costly
- objective evaluation is challenging
- a fragile concensus: BLEU [29]
  - measures the surface similarity with reference translation(s)
  - as the geometric mean of the n-gram precision
Get some numbers
Evaluating machine translation

- subjective evaluation is very costly
- objective evaluation is challenging
- a fragile consensus: BLEU [29]
  - measures the surface similarity with reference translation(s)
  - as the geometric mean of the n-gram precision

I am feeling good
Ref1: I am happy
Ref2: I am feeling very good

\[ p_1 = 1 \]
\[ p_2 = 2^{\frac{1}{3}} \]
\[ p_3 = 1^{\frac{1}{2}} \]
\[ p_4 = 0^{\frac{1}{1}} \]

an active research topic, many proposals are on the table
Get some numbers
Evaluating machine translation

- subjective evaluation is very costly
- objective evaluation is challenging
- a fragile consensus: BLEU [29]
  - measures the surface similarity with reference translation(s)
  - as the geometric mean of the \( n \)-gram precision
Get some numbers
Evaluating machine translation

- subjective evaluation is very costly
- objective evaluation is challenging
- a fragile consensus: BLEU [29]
  - measures the surface similarity with reference translation(s)
  - as the geometric mean of the \( n \)-gram precision
Get some numbers
Evaluating machine translation

- subjective evaluation is very costly
- objective evaluation is challenging
- a fragile consensus: **BLEU** [29]
  - measures the surface similarity with reference translation(s)
  - as the geometric mean of the \( n \)-gram precision
subjective evaluation is very costly
objective evaluation is challenging
a fragile consensus: BLEU [29]
- measures the surface similarity with reference translation(s)
- as the geometric mean of the $n$-gram precision

Ref1: I am happy
I am feeling good

Ref2: I am feeling very good
subjective evaluation is very costly

objective evaluation is challenging

a fragile consensus: BLEU [29]

measures the surface similarity with reference translation(s)

as the geometric mean of the $n$-gram precision

I am feeling good

Ref1: I am happy

Ref2: I am feeling very good

$p_1 = 1$
Get some numbers
Evaluating machine translation

- subjective evaluation is very costly
- objective evaluation is challenging
- a fragile consensus: BLEU [29]
  - measures the surface similarity with reference translation(s)
  - as the geometric mean of the \( n \)-gram precision

\[
\begin{align*}
I \text{ am feeling good} & \quad \text{Ref1: I am happy} \\
\text{Ref2: I am feeling very good} & \\
\end{align*}
\]

\[p_1 = 1 \quad p_2 = \frac{2}{3}\]
Get some numbers
Evaluating machine translation

- subjective evaluation is very costly
- objective evaluation is challenging
- a fragile consensus: **BLEU** [29]
  - measures the surface similarity with reference translation(s)
  - as the geometric mean of the \( n \)-gram precision

<table>
<thead>
<tr>
<th>Translation</th>
<th>Precision</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref1: I am happy</td>
<td>( p_1 = 1 )</td>
<td>Ref1: I am happy</td>
</tr>
<tr>
<td>I am feeling good</td>
<td>( p_2 = \frac{2}{3} )</td>
<td>Ref2: I am feeling very good</td>
</tr>
<tr>
<td>Ref2: I am feeling very good</td>
<td>( p_3 = \frac{1}{2} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( p_4 = 0 )</td>
<td></td>
</tr>
</tbody>
</table>

an active research topic, many proposals are on the table
Get some numbers
Evaluating machine translation

- subjective evaluation is very costly
- objective evaluation is challenging
- a fragile consensus: **BLEU** [29]
  - measures the surface similarity with reference translation(s)
  - as the geometric mean of the $n$-gram precision

```
I am feeling good
Ref1: I am happy
Ref2: I am feeling very good
```

\[ p_1 = 1 \quad p_2 = \frac{2}{3} \quad p_3 = \frac{1}{2} \quad p_4 = \frac{0}{1} \]

an active research topic, many proposals are on the table
A step back: finite-state SMT

- phrase-table lookup \([pt]\) is finite-state
- \(n\)-gram models \(lm\) can be implemented as weighted fSA
- monotonic decode of \(f\):
  \[ e^* = \text{bestpath}(\pi_2(f \circ pt) \circ lm) \] \[7\]
- decode with reordering
  \[ e^* = \text{bestpath}(\pi_2(\text{perm}(f) \circ pt) \circ lm) \] \[3\]
A step back: finite-state SMT

▶ phrase-table lookup \([pt]\) is finite-state
▶ \(n\)-gram models \(lm\) can be implemented as weighted fSA
▶ monotonic decode of \(f\):
  \[ e^* = \text{bestpath}(\pi_2(f \circ pt) \circ lm) \] \([7]\)
▶ decode with reordering
  \[ e^* = \text{bestpath}(\pi_2(\text{perm}(f) \circ pt) \circ lm) \] \([3]\)
A step back: finite-state SMT

- phrase-table lookup \([pt]\) is finite-state
- \(n\)-gram models \(lm\) can be implemented as weighted fSA
- monotonic decode of \(f\):
  \[ e^* = \text{bestpath}(\pi_2(f \circ pt) \circ lm) \] [7]
- decode with reordering
  \[ e^* = \text{bestpath}(\pi_2(\text{perm}(f) \circ pt) \circ lm) \] [3]
A step back: finite-state SMT

- phrase-table lookup \([pt]\) is finite-state
- \(n\)-gram models \(lm\) can be implemented as weighted fSA
- monotonic decode of \(f\):
  \[
e^* = \text{bestpath}(\pi_2(f \circ pt) \circ lm) \quad [7]
\]
- decode with reordering
  \[
e^* = \text{bestpath}(\pi_2(\text{perm}(f) \circ pt) \circ lm) \quad [3]
\]
A step back: finite-state SMT

- phrase-table lookup $[pt]$ is finite-state
- $n$-gram models $lm$ can be implemented as weighted fSA
- monotonic decode of $f$:
  $e^* = bestpath(\pi_2(f \circ pt) \circ lm)$ [7]
- decode with reordering
  $e^* = bestpath(\pi_2(\text{perm}(f) \circ pt) \circ lm)$ [3]

Efficient implementations, scalability, training procedures, non-deterministic input-outputs, integration of various knowledge-sources [18, 22]
A step back: finite-state SMT

- phrase-table lookup $[pt]$ is finite-state
- $n$-gram models $lm$ can be implemented as weighted fSA
- monotonic decode of $f$:
  \[ e^* = \text{bestpath}(\pi_2(f \circ pt) \circ lm) \] [7]
- decode with reordering
  \[ e^* = \text{bestpath}(\pi_2(\text{perm}(f) \circ pt) \circ lm) \] [3]

How to model $\text{perm}(f)$?
Approaches to reordering
Some attempts at modeling $perm(f)$

- **brute-force approach**
  + pruning based on distortion weights

- a priori defined permutations
  - define $T$, $perm(f) = f \circ T$
  - define $G$, $perm(f) = \{ f', S \Rightarrow (f; f') \}$

- empirically defined permutations
  - learn/train $T$, $perm(f) = f \circ T$
  - learn/train $T$, $perm(f) = \{ f', S \Rightarrow (f; f') \}$

- hand-crafted reordering rules

- any combination thereof

Small to mild gains with respect to monotonic translation; huge gap in performance between “easy” and “difficult” language pairs.
Approaches to reordering

Some attempts at modeling \( \text{perm}(f) \)

- \textbf{brute-force approach}
  - + pruning based on distortion weights

- \textbf{a priori defined permutations}
  - define \( T \), \( \text{perm}(f) = f \circ T \)
  - define \( G \), \( \text{perm}(f) = \{ f, S \Rightarrow (f; f') \} \)

- \textbf{empirically defined permutations}
  - learn/train \( T \), \( \text{perm}(f) = f \circ T \)
  - learn/train \( T \), \( \text{perm}(f) = \{ f', S \Rightarrow (f; f') \} \)

- \textbf{hand-crafted reordering rules}

- \textbf{any combination thereof}

Small to mild gains with respect to monotonic translation; huge gap in performance between “easy” and “difficult” language pairs.
Approaches to reordering

Some attempts at modeling $perm(f)$

- brute-force approach
  + pruning based on distortion weights

- a priori defined permutations
  - define $T$, $perm(f) = f \circ T$
  - define $G$, $perm(f) = \{f', S \Rightarrow^* (f; f')\}$

- empirically defined permutations
  - learn/train $T$, $perm(f) = f \circ T$
  - learn/train $T$, $perm(f) = \{f', S \Rightarrow (f; f')\}$

- hand-crafted reordering rules

- any combination thereof

Small to mild gains with respect to monotonic translation; huge gap in performance between “easy” and “difficult” language pairs.
Approaches to reordering

Some attempts at modeling $perm(f)$

- brute-force approach
  + pruning based on distortion weights

- a priori defined permutations
  - define $T$, $perm(f) = f \circ T$
  - define $G$, $perm(f) = \{f', S \Rightarrow^* (f; f')\}$

- empirically defined permutations
  - learn/train $T$, $perm(f) = f \circ T$
  - learn/train $T$, $perm(f) = \{f', S \Rightarrow (f; f')\}$

- hand-crafted reordering rules

- any combination thereof

Small to mild gains with respect to monotonic translation; huge gap in performance between “easy” and “difficult” language pairs.
Approaches to reordering

Some attempts at modeling $perm(f)$

- brute-force approach
  - try all permutations
  - + pruning based on distortion weights

- a priori defined permutations
  - define $T$, $perm(f) = f \circ T$
  - define $G$, $perm(f) = \{ f', S \Rightarrow^* (f; f') \}$

- empirically defined permutations
  - learn/train $T$, $perm(f) = f \circ T$
  - learn/train $T$, $perm(f) = \{ f', S \Rightarrow^* (f; f') \}$

- hand-crafted reordering rules
- any combination thereof

Small to mild gains with respect to monotonic translation; huge gap in performance between “easy” and “difficult” language pairs.
Approaches to reordering

Some attempts at modeling $perm(f)$

- **brute-force approach**
  - + pruning based on distortion weights

- **a priori defined permutations**
  - define $T$, $perm(f) = f \circ T$
  - define $G$, $perm(f) = \{f', S \Rightarrow^* (f; f')\}$

- **empirically defined permutations**
  - learn/train $T$, $perm(f) = f \circ T$
  - learn/train $T$, $perm(f) = \{f', S \Rightarrow^* (f; f')\}$

- **hand-crafted reordering rules**
- **any combination thereof**

Small to mild gains with respect to monotonic translation; huge gap in performance between “easy” and “difficult” language pairs.
Approaches to reordering
Some attempts at modeling $perm(f)$

- brute-force approach
  + pruning based on distortion weights

- a priori defined permutations
  - define $T$, $perm(f) = f \circ T$
  - define $G$, $perm(f) = \{ f', S \Rightarrow^* (f; f') \}$

- empirically defined permutations
  - learn/train $T$, $perm(f) = f \circ T$
  - learn/train $T$, $perm(f) = \{ f', S \Rightarrow^* (f; f') \}$

- hand-crafted reordering rules
- any combination thereof

Small to mild gains with respect to monotonic translation; huge gap in performance between “easy” and “difficult” language pairs.
Approaches to reordering
Some attempts at modeling $perm(f)$

- brute-force approach
  - try all permutations
  - + pruning based on distortion weights

- a priori defined permutations
  - define $T$, $perm(f) = f \circ T$
  - define $G$, $perm(f) = \{f', S \xrightarrow{*} (f; f')\}$

- empirically defined permutations
  - learn/train $T$, $perm(f) = f \circ T$
  - learn/train $T$, $perm(f) = \{f', S \xrightarrow{*} (f; f')\}$

- hand-crafted reordering rules
  - a man-made model

- any combination thereof

Small to mild gains with respect to monotonic translation; huge gap in performance between “easy” and “difficult” language pairs.
Approaches to reordering

Some attempts at modeling $\text{perm}(f)$

- brute-force approach
  + pruning based on distortion weights

- a priori defined permutations
  - define $T$, $\text{perm}(f) = f \circ T$
  - define $G$, $\text{perm}(f) = \{f', S \Rightarrow^* (f; f')\}$

- empirically defined permutations
  - learn/train $T$, $\text{perm}(f) = f \circ T$
  - learn/train $T$, $\text{perm}(f) = \{f', S \Rightarrow^* (f; f')\}$

- hand-crafted reordering rules

- any combination thereof

Small to mild gains with respect to monotonic translation; huge gap in performance between “easy” and “difficult” language pairs.
Approaches to reordering
Some attempts at modeling $perm(f)$

- brute-force approach
  + pruning based on distortion weights
- a priori defined permutations
  - define $T$, $perm(f) = f \circ T$
  - define $G$, $perm(f) = \{f', S \Rightarrow^* (f; f')\}$
- empirically defined permutations
  - learn/train $T$, $perm(f) = f \circ T$
  - learn/train $T$, $perm(f) = \{f', S \Rightarrow^* (f; f')\}$
- hand-crafted reordering rules
- any combination thereof

Small to mild gains with respect to monotonic translation; huge gap in performance between “easy” and “difficult” language pairs.
Approaches to reordering

Some attempts at modeling $perm(f)$

- brute-force approach
  + pruning based on distortion weights
- a priori defined permutations
  - define $T$, $perm(f) = f \circ T$
  - define $G$, $perm(f) = \{f', S \Rightarrow^* (f; f')\}$
- empirically defined permutations
  - learn/train $T$, $perm(f) = f \circ T$
  - learn/train $T$, $perm(f) = \{f', S \Rightarrow^* (f; f')\}$
- hand-crafted reordering rules
- any combination thereof

Small to mild gains with respect to monotonic translation; huge gap in performance between “easy” and “difficult” language pairs.
Approaches to reordering
Some attempts at modeling $perm(f)$

- brute-force approach
  + pruning based on distortion weights

- a priori defined permutations
  + define $T$, $perm(f) = f \circ T$
  + define $G$, $perm(f) = \{ f', S \Rightarrow^* (f; f') \}$

- empirically defined permutations
  + learn/train $T$, $perm(f) = f \circ T$
  + learn/train $T$, $perm(f) = \{ f', S \Rightarrow^* (f; f') \}$

- hand-crafted reordering rules

- any combination thereof

Small to mild gains with respect to monotonic translation; huge gap in performance between “easy” and “difficult” language pairs.
Why it works
PBT better than word based models

- idioms, terms, multi-word units
  - *pulling my leg, mène en bateau*

- “local” reordering decisions
  - *international conference, conférence internationale*

- model “local” context and agreement
  - *the international conference, la conférence internationale*

- allies simplicity, speed, and robustness

- matching large phrases yield high BLEU scores
Why it works
PBT better than word based models

- idioms, terms, multi-word units
  *pulling my leg, mène en bateau*

- “local” reordering decisions
  *international conference, conférence internationale*

- model “local” context and agreement
  *the international conference, la conférence internationale*

- allies simplicity, speed, and robustness

- matching large phrases yield high BLEU scores
Why it works
PBT better than word based models

► idioms, terms, multi-word units
   pulling my leg, mène en bateau

► “local” reordering decisions
   international conference, conférence internationale

► model “local” context and agreement
   the international conference, la conférence internationale

► allies simplicity, speed, and robustness

► matching large phrases yield high BLEU scores
Why it works
PBT better than word based models

- idioms, terms, multi-word units
  *pulling my leg, mènes en bateau*

- “local” reordering decisions
  *international conference, conférence internationale*

- model “local” context and agreement
  *the international conference, la conférence internationale*

- allies simplicity, speed, and robustness

- matching large phrases yield high BLEU scores
Why it works
PBT better than word based models

- idioms, terms, multi-word units
  *pulling my leg, mène en bateau*

- “local” reordering decisions
  *international conference, conférence internationale*

- model “local” context and agreement
  *the international conference, la conférence internationale*

- allies simplicity, speed, and robustness

- matching large phrases yield high BLEU scores
Why it works
PBT better than word based models

- idioms, terms, multi-word units
  *pulling my leg, mène en bateau*

- “local” reordering decisions
  *international conference, conférence internationale*

- model “local” context and agreement
  *the international conference, la conférence internationale*

- allies simplicity, speed, and robustness
- matching large phrases yield high BLEU scores
Why it works
PBT better than word based models

- idioms, terms, multi-word units
  *pulling my leg, mène en bateau*

- “local” reordering decisions
  *international conference, conférence internationale*

- model “local” context and agreement
  *the international conference, la conférence internationale*

- allies simplicity, speed, and robustness

- matching large phrases yield high BLEU scores
Why it works
PBT better than word based models

➤ idioms, terms, multi-word units
  *pulling my leg, mène en bateau*

➤ “local” reordering decisions
  *international conference, conférence internationale*

➤ model “local” context and agreement
  *the international conference, la conférence internationale*

➤ allies simplicity, speed, and robustness

➤ matching large phrases yield high BLEU scores
Why it fails
PBT worst than syntax-based models?

- purely surfacist (no morphology, see [19] for a cure)
- contiguous phrases miss important generalizations
- only “local” syntax on the target side (n-gram models)
- phrase weighting and selection is context-free
- no global reordering model
Why it fails

PBT worst than syntax-based models?

- purely surfacist (no morphology, see [19] for a cure)
- contiguous phrases miss important generalizations
- only “local” syntax on the target side (n-gram models)
- phrase weighting and selection is context-free
- no global reordering model
Why it fails

PBT worst than syntax-based models?

> purely surfacist (no morphology, see [19] for a cure)
> contiguous phrases miss important generalizations
> only “local” syntax on the target side ($n$-gram models)
> phrase weighting and selection is context-free
> no global reordering model
Why it fails
PBT worst than syntax-based models?

- purely surfacist (no morphology, see [19] for a cure)
- contiguous phrases miss important generalizations
- only “local” syntax on the target side (n-gram models)
- phrase weighting and selection is context-free
- no global reordering model
Why it fails
PBT worst than syntax-based models?

- purely surfacist (no morphology, see [19] for a cure)
- contiguous phrases miss important generalizations
- only “local” syntax on the target side (n-gram models)
- phrase weighting and selection is context-free
- no global reordering model
Temporary conclusions

- SMT’s recent progress = simpler models + larger databases + metrics
  - + tuning + paying attentions to details
  - acceptable translations for many pairs
  - issue: modeling word order ... with acceptable robustness and speed
  ⇒ towards more linguistically informed systems?
Temporary conclusions

- SMT’s recent progress = simpler models + larger databases + metrics
- + tuning + paying attentions to details
- acceptable translations for many pairs
- issue: modeling word order ... with acceptable robustness and speed
  ⇒ towards more linguistically informed systems?
Temporary conclusions

▶ SMT’s recent progress = simpler models + larger databases + metrics
▶ + tuning + paying attentions to details
▶ acceptable translations for many pairs
▶ issue: modeling word order ... with acceptable robustness and speed
⇒ towards more linguistically informed systems ?
Temporary conclusions

- SMT’s recent progress = simpler models + larger databases + metrics
- + tuning + paying attentions to details
- acceptable translations for many pairs
- issue: modeling word order ... with acceptable robustness and speed
  ⇒ towards more linguistically informed systems?
Questions ?
We have opened post-doctoral positions
Exhaustive search

- \( f \) has a finite number of permutations
- hence represented by a finite-state automaton
- yet can’t compute \( perm(f) \) with a finite-state device
Exhaustive search

- $f$ has a finite number of permutations
- hence represented by a finite-state automaton
- yet can’t compute $perm(f)$ with a finite-state device

Finite-state representation of $perm(123)$
Heuristic search

- moves allowed within fixed boundaries
- small moves preferred over longer moves
- standard model:
  - distortion: $d(i) = f(start(f_i) - end(f_{i-1}) - 1)$
  - $P(d(i) = k) \propto \exp(-\alpha k)$
  - $\forall i, d(i) < d_{\text{max}}$

- (costly) extension: lexicalized reordering weights [34]
IBM style constraints

- choose one the first $k$ remaining tokens

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>t=4</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>output = 0,2,3,5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t= 5</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>output = 0,2,3,5,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>⬤</td>
<td>o</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>output = 0,2,3,5,4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>⬤</td>
<td>o</td>
<td>⬤</td>
<td>⬤</td>
<td>o</td>
<td>⬤</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>current output 0,2,3,5,6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- additional constraints:
  - moves take place within a fixed size window;
  - restrict the number of simultaneous gaps;
IBM style constraints

- choose one the first $k$ remaining tokens
- additional constraints:
  - moves take place within a fixed size window;
  - restrict the number of simultaneous gaps;
IBM style constraints

- choose one of the first $k$ remaining tokens
- additional constraints:
  - moves take place within a fixed size window;
  - restrict the number of simultaneous gaps;

The IBM permutations of $a b c d$ for $k = 2$
A local approach

see [21] for details

- allows permutations of neighbouring phrases
- within a bounded window
A local approach
see [21] for details

- allows permutations of neighbouring phrases
- within a bounded window

One state $\forall a: A, b: B \in pt$, $???$ is a copy loop

Exchange adjacent phrases
A local approach

see [21] for details

- allows permutations of neighbouring phrases
- within a bounded window

5 states $\forall a:A, b:Bc:C \in pt, \ ?:?$ is a copy loop
Permute triplets of phrases
Definition (from [37])

An Inversion Transduction Grammar (ITG) is a 5-uple $G = (V, \Sigma, \Gamma, S, P)$, where the context-free productions:

- terminals come in pairs $a/b \in (\Sigma \cup \{\epsilon\}) \times (\Gamma \cup \{\epsilon\})$
- right-hand sides are explicitly oriented:
  - $A \rightarrow [BC]$: left-to-right order in both derivations
  - $A \rightarrow <BC>$: left-to-right in one language, right-to-left in the other
ITG’s permutations

Bracketing grammar
Let $G$ have productions $X \to [XX] | <XX>$, and $X \to e; e, \forall e$;
$perm(w_1 \ldots w_n) = \{ v_1 \ldots v_n \mid X \Rightarrow^* w_1 \ldots w_n; v_1 \ldots v_n \}$
ITG’s permutations

Bracketing grammar

Let $G$ have productions $X \to [XX] | < XX >$, and $X \to e; e, \forall e$;

$\text{perm}(w_1 \ldots w_n) = \{ v_1 \ldots v_n \mid X \Rightarrow^* w_1 \ldots w_n; v_1 \ldots v_n \}$
ITG’s permutations

Bracketing grammar

Let $G$ have productions $X \rightarrow [XX] | <XX>$, and $X \rightarrow e; e, \forall e$;

$\text{perm}(w_1 \ldots w_n) = \{v_1 \ldots v_n \mid X \Rightarrow^* w_1 \ldots w_n; v_1 \ldots v_n\}$
ITG’s permutations

Bracketing grammar

Let $G$ have productions $X \rightarrow [XX] | <XX>$, and $X \rightarrow e; e, \forall e$;

$\text{perm}(w_1 \ldots w_n) = \{v_1 \ldots v_n \mid X \Rightarrow^* w_1\ldots w_n; v_1\ldots v_n\}$
ITG’s permutations

Bracketing grammar

Let $G$ have productions $X \rightarrow [XX] | <XX>$, and $X \rightarrow e; e, \forall e$;

$perm(w_1 \ldots w_n) = \{ v_1 \ldots v_n \mid X \Rightarrow^* w_1 \ldots w_n; v_1 \ldots v_n \}$
ITG’s permutations

Bracketing grammar
Let $G$ have productions $X \rightarrow [XX] \mid <XX>$, and $X \rightarrow e; e, \forall e$;
$\text{perm}(w_1 \ldots w_n) = \{v_1 \ldots v_n \mid X \Rightarrow^* w_1 \ldots w_n; v_1 \ldots v_n\}$

Complements
- a strict subset of all permutations
- combinatorially large $O(K^n)$ [39], yet $\ll n!$
- can be searched in polynomial time [39, 14]
Linguistic reordering

- use linguistically motivated transformations rules eg. [11]

> **Verb Initial Rule**

*In any verb phrase, find the head of the phrase, and move it into the initial position within the verb phrase*

\[f = \text{Ich werde Ihnen die entsprechenden Anmerkungen aushändigen}
\]

\[f' = \text{Ich werde aushändigen ihnen die entsprechenden Anmerkungen}
\]

\[e = \text{I will pass on to you the corresponding comments}
\]

- deterministic process \(\Rightarrow\) transform dataset prior to learning
- requirements: a source parser + linguistic rules (for each pair)
Learning reordering rules
see eg. [38, 12]

- **training procedure**
  - build symmetric alignments and extract phrases
  - learn “within-phrase” reordering rules
  - compose rules as a non-deterministic reordering transducer $R$

$$R = \bigcirc_i (r_i \cup \text{Id})$$

- decoding uses $\text{perm}(f) = \pi_1 (\text{tag}(f) \circ R)$
Learning reordering rules

see eg. [38, 12]

- training procedure
  - build symmetric alignments and extract phrases
  - learn “within-phrase” reordering rules
  - compose rules as a non-deterministic reordering transducer $R$

\[
R = \bigcirc_i (r_i \cup \text{Id})
\]

- decoding uses $\text{perm}(f) = \pi_1 (\text{tag}(f) \circ R)$

ideas: $NN$

excellent:

excellentes: $JJ$

and:

y: $CC$

constructivas: $JJ$

constructive:

ideas:
Learning reordering rules

see eg. [38, 12]

▶ training procedure
  ▶ build symmetric alignments and extract phrases
  ▶ learn “within-phrase” reordering rules
  ▶ compose rules as a non-deterministic reordering transducer \( R \)
    \[
    R = \bigcirc_i (r_i \cup \text{Id})
    \]

▶ decoding uses \( \text{perm}(f) = \pi_1 (\text{tag}(f) \circ R) \)

excellent and constructive ideas:

excellentes: \(JJ\)  
and

constructivas: \(JJ\)  
ideas: \(NN\)

rule: \(NN \ JJ \ CC \ JJ \rightarrow JJ \ CC \ JJ \ NN\)
Learning reordering rules
see eg. [38, 12]

▶ training procedure
  ▶ build symmetric alignments and extract phrases
  ▶ learn “within-phrase” reordering rules
  ▶ compose rules as a non-deterministic reordering transducer $R$

\[
R = \bigcirc_i (r_i \cup \text{Id})
\]

▶ decoding uses $perm(f) = \pi_1 (tag(f) \circ R)$
Learning reordering rules
see eg. [38, 12]

- training procedure
  - build symmetric alignments and extract phrases
  - learn “within-phrase” reordering rules
  - compose rules as a non-deterministic reordering transducer $R$
    \[
    R = \bigcirc_i (r_i \cup \text{Id})
    \]

- decoding uses $\text{perm}(f) = \pi_1 (\text{tag}(f) \circ R)$
Extracting gappy phrases

When translating from French to English, we can use gappy phrases to capture the meaning more accurately. For example, the phrase "tu ne veux pas dormir" can be translated as "you don’t want to sleep." The French phrase contains a gappy phrase of "don’t want; ne veux pas".

This gappy phrase can be generalized to "N(don’t X; ne X pas)++", which allows for better generalization in translation.

Here is a table to illustrate the relationship between the words in French and their English translations:

<table>
<thead>
<tr>
<th></th>
<th>tu</th>
<th>ne</th>
<th>veux</th>
<th>pas</th>
<th>dormir</th>
</tr>
</thead>
<tbody>
<tr>
<td>sleep to want don’t you</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- (want; veux) a sub-phrase of (don’t want; ne veux pas)
- ⇒ gappy phrase N(don’t X; ne X pas)++
- better generalization
Extracting gappy phrases

\[ f = \text{je ne le comprends plus} \]
\[ e = \text{I don’t understand it anymore} \]

- same idea, with two variables
- \[ N(\text{don’t } X_1 X_2 \text{ anymore } ; \text{ ne } X_2 X_1 \text{ plus}) \]
- defines a (lexicalized) reordering model
A hierarchical SMT system

Some innovations of [9]

- gappy phrases = rules of a synchronous CFG
  - usual phrases \((e; f)\) yield terminating rules \(X \rightarrow e; f\)
  - gappy phrases \((\alpha; \beta)\) yield \(X \rightarrow \alpha; \beta\)
  - “glue” \(S \rightarrow SX \mid X\)
  - maximum likelihood estimates (+ smoothing)

- translation within parsing

\[
e = \arg\max_{e \in E} \lambda_1 \log P_{LM}(e) + \lambda_2 \log P_G(f; e) + \ldots
\]

- Benefits
  - more (general) phrases
  - reordering model
  - performance [41]

- Issues
  - grammar size
  - search
Translating with Stochastic Synchronous CFGs

- in theory:

\[ e^* = \arg \max_{e \in E} P_G(f : e) = \sum_{\{(t, t'), \text{yield}(t) = f \land \text{yield}(t') = e\}} P_G(t, t') \]

this problem is intractable (even in one dimension see [31])

- a two step approach
  1. solve

\[ t^* = \arg \max \left[P_G(t) = \sum_{t'} P_G(t, t')\right] \quad \text{for} \quad \{t | \text{yield}(t) = f\} \]

  2. derive \( e^* \) as

\[ e^* = \arg \max_e \sum_{\{t' | \text{yield}(t') = e\}} P_G(t^*, t') \]

the first step is also NP-hard [30]

- a polynomial (?) approximation

\[ e^* = \text{yield}(t^*), \text{where} (t, t^*) = \arg \max_{\{t | \text{yield}(t) = f\}, t'} P_G(t, t') \]


References II


References IV


This beautiful plant is unique

Courtesy of Ph. Langlais

<table>
<thead>
<tr>
<th>phrase table</th>
</tr>
</thead>
<tbody>
<tr>
<td>this ↔ ce</td>
</tr>
<tr>
<td>↔ cette</td>
</tr>
<tr>
<td>beautiful ↔ belle</td>
</tr>
<tr>
<td>↔ beau</td>
</tr>
<tr>
<td>plant ↔ plante</td>
</tr>
<tr>
<td>↔ usine</td>
</tr>
<tr>
<td>is ↔ est</td>
</tr>
<tr>
<td>unique ↔ seule</td>
</tr>
<tr>
<td>↔ unique</td>
</tr>
<tr>
<td>beautiful plant</td>
</tr>
<tr>
<td>↑</td>
</tr>
<tr>
<td>belle plante</td>
</tr>
<tr>
<td>plante magnifique</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>language model</th>
</tr>
</thead>
</table>
| ce beau plante | :-(
| cette belle usine | :-|
| belle usine est | :-) |
| ...            |
This beautiful plant is unique
Courtesy of Ph. Langlais

### phrase table

<table>
<thead>
<tr>
<th>English</th>
<th>French</th>
</tr>
</thead>
<tbody>
<tr>
<td>this</td>
<td>ce</td>
</tr>
<tr>
<td>beautiful</td>
<td>belle</td>
</tr>
<tr>
<td>plant</td>
<td>plante</td>
</tr>
<tr>
<td>is</td>
<td>est</td>
</tr>
<tr>
<td>unique</td>
<td>seule</td>
</tr>
</tbody>
</table>

### language model

- ce beau plante :-(
- cette belle usine :-(
- belle usine est :-)

...
This beautiful plant is unique
Courtesy of Ph. Langlais
This beautiful plant is unique
Courtesy of Ph. Langlais
This beautiful plant is unique
Courtesy of Ph. Langlais

phrase table

<table>
<thead>
<tr>
<th>English</th>
<th>French</th>
</tr>
</thead>
<tbody>
<tr>
<td>this</td>
<td>ce</td>
</tr>
<tr>
<td>beautiful</td>
<td>belle</td>
</tr>
<tr>
<td>plant</td>
<td>plante</td>
</tr>
<tr>
<td>is</td>
<td>est</td>
</tr>
<tr>
<td>unique</td>
<td>seule</td>
</tr>
</tbody>
</table>

language model

ce beau plante  :-(
Cette belle usine  ::|
belle usine est  :-)

...
This beautiful plant is unique
Courtesy of Ph. Langlais
This beautiful plant is unique

Courtesy of Ph. Langlais
A finite-state representation of a phrase-table
A second step back

Abstract SMT

1. get weighted local translation hypotheses from the PT
2. arrange them in a word graph
3. rescore permutations with a language model

Two steps forward

- compute weights *on demand*, using all available information: SMT as EBMT [32], see also [35, 23]
- dispense with alignments in step 1, use complete sentence as contexts
  (but step 2 and 3 prove difficult [2])
A second step back

Abstract SMT

1. get weighted local translation hypotheses from the PT
2. arrange them in a word graph
3. rescore permutations with a language model

Two steps forward

- compute weights *on demand*, using all available information: SMT as EBMT [32], see also [35, 23]
- dispense with alignments in step 1, use complete sentence as contexts
  (but step 2 and 3 prove difficult [2])
Using Terascale Language Models

Some results from [5]

Conventional back-off

\[ P(w|h) = \begin{cases} 
\rho(hw) & \text{if } N(hw) > 0 \\
\alpha(h)P(w|h) & \text{otherwise} 
\end{cases} \]

"Stupid" (sic) Back-off

\[ S(w|h) = \begin{cases} 
\frac{N(hw)}{\sum_{w'} N(hw')} & \text{if } N(hw) > 0 \\
\alpha S(W|h) & \text{otherwise} 
\end{cases} \]

NB. “Stupid” Back-off does not even define a probability distribution
Using Terascale Language Models

Some results from [5]

<table>
<thead>
<tr>
<th></th>
<th>target</th>
<th>webnews</th>
<th>web</th>
</tr>
</thead>
<tbody>
<tr>
<td># token</td>
<td>237 M</td>
<td>31G</td>
<td>1.8T</td>
</tr>
<tr>
<td>vocab size</td>
<td>200k</td>
<td>5M</td>
<td>16M</td>
</tr>
<tr>
<td># ngrams size (B)</td>
<td>257M</td>
<td>21 G</td>
<td>300G</td>
</tr>
<tr>
<td>size (B)</td>
<td>2G</td>
<td>89G</td>
<td>1.8 T</td>
</tr>
<tr>
<td>time (SB)</td>
<td>20 min</td>
<td>8 hours</td>
<td>1 day</td>
</tr>
<tr>
<td>time (KN)</td>
<td>2.5 hours</td>
<td>2 days</td>
<td>-</td>
</tr>
</tbody>
</table>
Using Terascale Language Models

Some results from [5]
A real world phrase-table
Based on the en-fr Europarl

467 (en → fr) translations for “European Commission”

European Commission ||| Commission européenne
European Commission ||| Commission
European Commission ||| la Commission européenne
European Commission ||| Commission européenne ,
European Commission ||| de la Commission européenne
(...)

98 (fr → en) translations for “cultures”

cultures ||| agriculture
cultures ||| arable
cultures ||| crop production
cultures ||| cultivation
cultures ||| cultural content
cultures ||| cultural history
cultures ||| drug crops
cultures ||| farming
cultures ||| farms
cultures ||| identities
cultures ||| language
cultures ||| plants
(...)

(...)

back
A real world phrase-table
Based on the en-fr Europarl

467 (en → fr) translations for “European Commission”

<table>
<thead>
<tr>
<th>English</th>
<th>French</th>
</tr>
</thead>
<tbody>
<tr>
<td>European Commission</td>
<td>Commission européenne</td>
</tr>
<tr>
<td>European Commission</td>
<td>Commission</td>
</tr>
<tr>
<td>European Commission</td>
<td>la Commission européenne</td>
</tr>
<tr>
<td>European Commission</td>
<td>Commission européenne ,</td>
</tr>
<tr>
<td>European Commission</td>
<td>de la Commission européenne</td>
</tr>
</tbody>
</table>

(...)

98 (fr → en) translations for “cultures”

<table>
<thead>
<tr>
<th>French</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>cultures</td>
<td>agriculture</td>
</tr>
<tr>
<td>cultures</td>
<td>arable</td>
</tr>
<tr>
<td>cultures</td>
<td>crop production</td>
</tr>
<tr>
<td>cultures</td>
<td>cultivation</td>
</tr>
<tr>
<td>cultures</td>
<td>cultural content</td>
</tr>
<tr>
<td>cultures</td>
<td>cultural history</td>
</tr>
<tr>
<td>cultures</td>
<td>drug crops</td>
</tr>
<tr>
<td>cultures</td>
<td>farming</td>
</tr>
<tr>
<td>cultures</td>
<td>farms</td>
</tr>
<tr>
<td>cultures</td>
<td>identities</td>
</tr>
<tr>
<td>cultures</td>
<td>language</td>
</tr>
<tr>
<td>cultures</td>
<td>plants</td>
</tr>
</tbody>
</table>

(...)

(...)

back
A real world phrase-table
Based on the en-fr Europarl

672 translations for ’!’ !!!

! ||| ! ! !
! ||| ! !
! ||| ! |||
! ||| : non !
...
! ||| , dit-on partout !
! ||| , exigez que
! ||| , exigez
! ||| , il est primordial que la
! ||| , il est primordial que
...
! ||| Messieurs , il est primordial que la
! ||| Messieurs , il est primordial
...

▶ back
mais là-dessus je voudrais marquer sinon un désaccord, du moins des nuances sur deux points.

but I would like to indicate otherwise a disagreement, at least the nuances on two points

From Europarl 2008
n’ y a -t-il pas ici deux poids , deux mesures ?

is there not here two weights , two measures ?

From Europarl 2008
en réalité, les entrepreneurs sont plus souvent comparables à des joueurs qui espèrent toucher le pactole.

in reality, the entrepreneurs are more often comparable to players who are hoping to touch the gold mine.

From Europarl 2008
les investisseurs plus vigilants achetent déjà en grand nombre, par exemple dans le coin de Bansko.

investors more vigilant achetent already in great numbers, for example in the corner of Bansko.

From NewsTest 2008
l’ avocat des familles sinistrées Igor Veleba veut obtenir de l’hôpital de Motol un dédommagement de 12 millions de couronnes plus les dépens.

the lawyer of Igor Veleba affected families to obtain the hospital Motol compensation of 12 million kronor more expense.

From NewsTest 2008