Confidence-based Rewriting of Machine Translation Output

Benjamin Marie\textsuperscript{1,2} \hspace{1cm} Aurélien Max\textsuperscript{1,3}

(1) LIMSI-CNRS \hspace{1cm} (2) Lingua et Machina \hspace{1cm} (3) Université Paris-Sud
Introduction

- Phrase-Based Statistical Machine Translation (PBSMT) systems use many features during decoding to assess the quality of translation hypotheses.

- For other features, several difficulties of integration to overcome, e.g.:
  - **need of a complete hypothesis**
    - e.g. sentence-level syntactic features
  - **computational cost**
    - e.g. Neural Network language models
  - **need of a first decoding**
    - e.g. *a posteriori* confidence models

- How to use such features *efficiently* in PBSMT?
Reranking of translation hypotheses

A solution

- rerank the $n$-best list of the decoder using new, complex features
- can achieve good performance with some features

(Och et al., 2004; Carter and Monz, 2011; Le et al., 2012; Luong et al., 2014)

2 strong limitations

- lack of diversity (Gimpel et al., 2013)
- inherit a limited selection of hypotheses made by the decoder
A rewriting system
A rewriter to extend the exploration

- idea: search for new promising hypotheses **not** in the \( n \)-best list
The seed: an hypothesis to rewrite
A rewriting phrase table

rewriting phrase table

seed
A set of rewriting operations

rewriting phrase table

operations
replace
merge
split

seed
Neighborhood generation

- Seed
- Generate neighborhood
- Rewriting phrase table
- Operations: replace, merge, split
Neighborhood generation: replace

- French: "il a refusé le test immédiatement.
- English: "he has refused a test now."
Neighborhood generation: replace

*il a refusé le test immédiatement.*

*he has refused a test now.*
Neighborhood generation: replace

il a refusé le test immédiatement.

he has refused a test now.

he refused a test now.

he had refused a test now.

it has refused a test now.

it refused a test now.
Neighborhood generation: split

"il a refusé le test immédiatement." → "he has refused a test now."
Neighborhood generation: *split*

```
il a refusé le test immédiatement.
he has refused a test now.
```
Neighborhood generation: split

il a refusé le test immédiatement.

he has refused a test now.

he is refused a test now.

he had refused a test now.

it has refused a test now.

it have refused a test now.
Neighborhood generation: merge

\[
\text{il a refusé \ le test \ immédiatement.} \\
\text{he has refused \ a test \ now.}
\]
Neighborhood generation: merge

*il a refusé le test immédiatement.*

—he has refused a test now.
Neighborhood generation: merge

- il a refusé le test immédiatement.
- he has refused a test now.
- he refused a test now.
- he rejected a test now.
- he has just refused a test now.
- he has a test now.
Rewriting phrase table

Building the rewriting table

- **Method 1**: take the $i$ best translations according to $p(e|f)$
- **Method 2**: take the bi-phrases appearing in the decoder $k$-best list

**Method 1**
- produces very large neighborhoods
- not suitable for costly features

**Method 2**
- produces very small and adapted rewriting phrase table for each sentence
- keeps only bi-phrases for which the decoder was the most confident
Neighborhood generation

- Seed
- Generate neighborhood
- Rewriting phrase table
- Operations
  - Replace
  - Merge
  - Split
Ranking of the neighborhood

- rewriting phrase table
- operations: replace, merge, split
- seed
- generate neighborhood
- rank neighborhood

1 hypothesis
2 hypothesis
3 hypothesis
4 hypothesis
\vdots
i hypothesis
Ranking of the neighborhood

Objective

- rank (manageable) neighborhoods using complex features

Training the reranker: 2 kinds of examples

- $n$-best produced by the decoder
- neighborhoods produced by one iteration of rewriter

Training algorithm

- kb-mira (Cherry and Foster, 2012)
Ranking of the neighborhood

- Rewriting phrase table
- Operations: replace, merge, split
- Seed
- Generate neighborhood
- Rank neighborhood
- 1 hypothesis
- 2 hypothesis
- 3 hypothesis
- 4 hypothesis
- ..
- i hypothesis
Greedy search

- **seed**
- **generate neighborhood**
- **neighborhood**
- **operations**
  - replace
  - merge
  - split
- **rewriting phrase table**
- **1 hypothesis**
- **2 hypothesis**
- **3 hypothesis**
- **4 hypothesis**
- **i hypothesis**
- **rank neighborhood**
- **1-best == seed**
- return 1-best

Benjamin MARIE (LIMSI-CNRS)

Confidence-based Rewriting of MT output

10/2014 24 / 47
Greedy search

1-best == seed
1-best != seed

return 1-best

seed

generate neighborhood

neighborhood

rewriting phrase table

operations

replace
merge
split

rank neighborhood

1 hypothesis
2 hypothesis
3 hypothesis
4 hypothesis
i hypothesis

seed ← 1-best
Greedy search

- greedy search algorithm for PBSMT (Langlais et al., 2007)
  - choose at each iteration the best rewriting/operation according to the (new) scoring function

<table>
<thead>
<tr>
<th>Source</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>il a refusé le test immédiatement.</td>
<td>he refused the test straight away.</td>
</tr>
</tbody>
</table>

**seed**

<table>
<thead>
<tr>
<th>Source</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>il a$_1$ refusé$_2$ le test$_3$ immédiatement .$_4$</td>
<td>he has$_1$ refused$_2$ a test$_3$ now .$_4$</td>
</tr>
</tbody>
</table>
Greedy search

- greedy search algorithm for PBSMT \cite{Langlais2007} (Langlais et al., 2007)
  - choose at each iteration the best rewriting/operation according to the (new) scoring function

<table>
<thead>
<tr>
<th>Source</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>il a refusé le test immédiatement .</td>
<td>he refused the test straight away .</td>
</tr>
</tbody>
</table>

**seed**

- il a refusé\textsubscript{1} le test\textsubscript{3} immédiatement .4
- he has\textsubscript{1} refused\textsubscript{2} a test\textsubscript{3} now .4

**merge**

- il a refusé\textsubscript{1} le test\textsubscript{2} immédiatement .3
- he refused\textsubscript{1} a test\textsubscript{2} now .3

**iteration 1**
Greedy search

- greedy search algorithm for PBSMT (Langlais et al., 2007)
  - choose at each iteration the best rewriting/operation according to the (new) scoring function

### Source

<table>
<thead>
<tr>
<th>Source</th>
<th>il a refusé le test immédiatement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>he refused the test straight away.</td>
</tr>
<tr>
<td><strong>seed</strong></td>
<td>il a₁ refusé₂ le test₃ immédiatement₄</td>
</tr>
<tr>
<td>↓</td>
<td>he has₁ refused₂ a test₃ now₄</td>
</tr>
<tr>
<td><strong>merge</strong></td>
<td>il a refusé₁ le test₂ immédiatement₃</td>
</tr>
<tr>
<td><strong>iteration 1</strong></td>
<td>he refused₁ a test₂ now₃</td>
</tr>
<tr>
<td><strong>split</strong></td>
<td>il a refusé₁ le test₂ immédiatement₃</td>
</tr>
<tr>
<td><strong>iteration 2</strong></td>
<td>he refused₁ a test₂ straight away₃</td>
</tr>
</tbody>
</table>

Benjamin MARIE (LIMSI-CNRS)
Greedy search

- greedy search algorithm for PBSMT (Langlais et al., 2007)
  - choose at each iteration the best rewriting/operation according to the (new) scoring function

<table>
<thead>
<tr>
<th>Source</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>il a refusé le test immédiatement .</td>
<td>he refused the test straight away .</td>
</tr>
</tbody>
</table>

**seed**
- il a refusé_1 le test_2 immédiatement .
- he has_1 refused_2 a test_3 now .4

**merge**
- il a refusé_1 le test_2 immédiatement .

**iteration 1**
- he refused_1 a test_2 now .3

**split**
- il a refusé_1 le test_2 immédiatement .

**iteration 2**
- he refused_1 a test_2 straight away .4

**replace**
- il a refusé_1 le test_2 immédiatement .

**iteration 3**
- he refused_1 the test_2 straight away .4
Experiments
The whole framework
Experimental settings

- **translation tasks**: English↔French
  - Ted Talks
  - WMT’14 medical
  - WMT’12

- **baseline systems**
  - Moses PBSMT (Koehn et al., 2007)
  - kb-mira reranker using all the features below

- **features**
  - decoder features: all the features used by the 1st-pass decoder
  - neural network models: 10-gram monolingual (Le et al., 2011) and bilingual (Le et al., 2012) SOUL models
  - Part-of-speech language model: 6-gram model
  - IBM1 scores
  - phrase posterior probabilities
## Results

<table>
<thead>
<tr>
<th>Task</th>
<th>system</th>
<th>en-fr BLEU</th>
<th>en-fr Δ</th>
<th>fr-en BLEU</th>
<th>fr-en Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMT’12</td>
<td>1-pass Moses</td>
<td>31.8</td>
<td></td>
<td>29.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>reranker</td>
<td>32.9</td>
<td>+1.1</td>
<td>30.3</td>
<td>+0.9</td>
</tr>
<tr>
<td>TED Talks</td>
<td>1-pass Moses</td>
<td>32.3</td>
<td></td>
<td>32.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>reranker</td>
<td>32.8</td>
<td>+0.5</td>
<td>33.0</td>
<td>+0.5</td>
</tr>
<tr>
<td>WMT’14 medical</td>
<td>1-pass Moses</td>
<td>38.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>reranker</td>
<td>41.8</td>
<td>+3.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

⇒ moderate (TED Talks) to strong (medical) improvements with reranker over the 1st-pass decoder
## Results

<table>
<thead>
<tr>
<th>Task</th>
<th>system</th>
<th>en-fr</th>
<th>fr-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMT’12</td>
<td>1-pass Moses</td>
<td>31.8</td>
<td>29.4</td>
</tr>
<tr>
<td></td>
<td>reranker</td>
<td>32.9</td>
<td>30.3</td>
</tr>
<tr>
<td></td>
<td>rewriter</td>
<td>33.5</td>
<td>30.8</td>
</tr>
<tr>
<td>TED Talks</td>
<td>1-pass Moses</td>
<td>32.3</td>
<td>32.5</td>
</tr>
<tr>
<td></td>
<td>reranker</td>
<td>32.8</td>
<td>33.0</td>
</tr>
<tr>
<td></td>
<td>rewriter</td>
<td>33.7</td>
<td>33.4</td>
</tr>
<tr>
<td>WMT’14 medical</td>
<td>1-pass Moses</td>
<td>38.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>reranker</td>
<td>41.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rewriter</td>
<td>43.4</td>
<td></td>
</tr>
</tbody>
</table>

⇒ **rewriter increases by ∼50% the reranker improvement**
## Results

<table>
<thead>
<tr>
<th>Task</th>
<th>system</th>
<th>en-fr BLEU</th>
<th>fr-en BLEU</th>
<th>∆ en-fr</th>
<th>∆ fr-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMT’12</td>
<td>1-pass Moses</td>
<td>31.8</td>
<td>29.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>reranker</td>
<td>32.9</td>
<td>30.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rewriter</td>
<td>33.5 +0.6</td>
<td>30.8 +0.5</td>
<td>+0.6</td>
<td>+0.5</td>
</tr>
<tr>
<td>TED Talks</td>
<td>1-pass Moses</td>
<td>32.3</td>
<td>32.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>reranker</td>
<td>32.8</td>
<td>33.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rewriter</td>
<td>33.7 +0.9</td>
<td>33.4 +0.4</td>
<td>+0.9</td>
<td>+0.4</td>
</tr>
<tr>
<td>WMT’14 medical</td>
<td>1-pass Moses</td>
<td>38.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>reranker</td>
<td>41.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rewriter</td>
<td>43.4 +1.6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

⇒ rewriter increases by ~50% the reranker improvement
Analysis: outline

1. training procedure
2. rewriting phrase table
3. best attainable performance
4. performance depending on translation quality
5. sentence-level performance
6. other findings
## Training examples

<table>
<thead>
<tr>
<th>Method</th>
<th>dev BLEU</th>
<th>test BLEU</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>reranker</td>
<td>44.1</td>
<td>41.8</td>
<td></td>
</tr>
<tr>
<td>rewriter training</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-pass Moses 1,000-best</td>
<td>44.1</td>
<td>39.2</td>
<td>-2.6</td>
</tr>
<tr>
<td>rewriter neighborhoods</td>
<td>44.5</td>
<td>43.4</td>
<td>+1.6</td>
</tr>
</tbody>
</table>

⇒ rewriter **must** be trained on rewriter neighborhoods
Rewriting phrase table performance

Method 1: extraction according to $p(e|f)$
  - damages reranker output

Method 2: extraction from a $k$-best list
  - improvements for all tested $k$, even for small values (best for $k = 10,000$)
Rewriting phrase table performance

Method 1: extraction according to $p(e|f)$
  - damages reranker output

Method 2: extraction from a $k$-best list
  - improvements for all tested $k$, even for small values (best for $k = 10,000$)
Rewriting phrase table performance

Method 1: extraction according to $p(e|f)$
  - damages reranker output

Method 2: extraction from a $k$-best list
  - improvements for all tested $k$, even for small values (best for $k = 10,000$)
## Rewriting phrase table size

<table>
<thead>
<tr>
<th>Rewriting phrase table</th>
<th>Unique bi-phrases</th>
<th>Δ-BLEU w.r.t. reranker</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Method 1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$i = 5$</td>
<td>85,530</td>
<td>-0.8</td>
</tr>
<tr>
<td>$i = 10$</td>
<td>149,887</td>
<td>-0.7</td>
</tr>
<tr>
<td><strong>Method 2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k = 10$</td>
<td>21,398</td>
<td>+0.6</td>
</tr>
<tr>
<td>$k = 100$</td>
<td>28,730</td>
<td>+1.1</td>
</tr>
<tr>
<td>$k = 1,000$</td>
<td>33,929</td>
<td>+1.2</td>
</tr>
<tr>
<td>$k = 10,000$</td>
<td>38,455</td>
<td>+1.6</td>
</tr>
</tbody>
</table>

- compact phrase tables when extracted from $k$-best lists (Method 2)
- much larger when extracted according to $p(e|f)$ (Method 1)
Best attainable performance

- Greedy Oracle Search (GOS) *(Marie and Max, 2013)*
  - make the best local decision at each iteration
  - use sentence-BLEU as scoring function

<table>
<thead>
<tr>
<th>baseline reranker</th>
<th>test BLEU</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>reranker</td>
<td>41.8</td>
<td></td>
</tr>
<tr>
<td>rewriting phrase table</td>
<td></td>
<td></td>
</tr>
<tr>
<td>method 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( i = 5 )</td>
<td>50.6</td>
<td>+8.8</td>
</tr>
<tr>
<td>( i = 10 )</td>
<td>54.5</td>
<td>+12.7</td>
</tr>
<tr>
<td>method 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( k = 10 )</td>
<td>45.9</td>
<td>+4.1</td>
</tr>
<tr>
<td>( k = 100 )</td>
<td>50.2</td>
<td>+8.4</td>
</tr>
<tr>
<td>( k = 1,000 )</td>
<td>53.3</td>
<td>+11.5</td>
</tr>
<tr>
<td>( k = 10,000 )</td>
<td><strong>58.7</strong></td>
<td>+16.9</td>
</tr>
</tbody>
</table>

⇒ strong oracle improvements, even for compact rewriting tables
⇒ extracting from \( k \)-best lists much more promising
Performance depending on translation quality

- **rewriter improvement**:
  - quartile 4: +1.4 BLEU
  - quartile 1: +9.0 BLEU

⇒ larger improvements on bad/difficult translations
Sentence-level performance

- according to sentence-BLEU, after rewriting:
  - 40.8% better
  - 29.2% worse
  - 30% unchanged

⇒ large room for further improvement
Sentence-level performance: semi-oracle experiment

- (a) automatic rewriting
- (b) semi-oracle rewriting

- Protecting the phrases appearing in the reference translation: +1.5 BLEU
- Strong value of better confidence estimates
Other findings

1. 70% of new hypotheses not in 1-pass Moses 1,000-best
2. on average (only) 116 hypotheses per sentence in the neighborhood
3. searching using a beam of size 10: 1.6 $\rightarrow$ 1.9 BLEU
4. manual evaluation revealed both fluency and accuracy improvements
Conclusion

- an **efficient** and **simple** procedure to make a **better use of features**
difficult to integrate during decoding

- produces useful hypotheses not in the decoder $n$-best list

- relies on the **decoder confidence** to extract the rewriting rules

- improvements on **3 different tasks** and **2 language directions** over a reranked baseline using the same features
Future work

- exploit more features: lexical-coherence (Hardmeier et al., 2012), syntactic features (Post, 2011), word posterior probability (Ueffing and Ney, 2007), etc.

- identify correct phrases to protect them from rewriting

- adapt rewriter’s objective function to the sentence

- use a paraphrase operation rewriting the source sentence to produce new target phrases (Marie and Max, 2013)

- use automatic alternative reference translations (Madnani and Dorr, 2013)

- use rewriter in interaction with human translators
Future work

- exploit more features: lexical-coherence (Hardmeier et al., 2012), syntactic features (Post, 2011), word posterior probability (Ueffing and Ney, 2007), etc.
- identify correct phrases to protect them from rewriting
- adapt rewriter’s objective function to the sentence
- use a paraphrase operation rewriting the source sentence to produce new target phrases (Marie and Max, 2013)
- use automatic alternative reference translations (Madnani and Dorr, 2013)
- use rewriter in interaction with human translators
Future work

- exploit more features: lexical-coherence (Hardmeier et al., 2012), syntactic features (Post, 2011), word posterior probability (Ueffing and Ney, 2007), etc.

- identify correct phrases to protect them from rewriting

- adapt rewriter’s objective function to the sentence

- use a paraphrase operation rewriting the source sentence to produce new target phrases (Marie and Max, 2013)

- use automatic alternative reference translations (Madnani and Dorr, 2013)

- use rewriter in interaction with human translators
Future work

▶ exploit more features: lexical-coherence (Hardmeier et al., 2012), syntactic features (Post, 2011), word posterior probability (Ueffing and Ney, 2007), etc.

▶ identify correct phrases to protect them from rewriting

▶ adapt rewriter’s objective function to the sentence

▶ use a paraphrase operation rewriting the source sentence to produce new target phrases (Marie and Max, 2013)

▶ use automatic alternative reference translations (Madnani and Dorr, 2013)

▶ use rewriter in interaction with human translators
Future work

- exploit more features: lexical-coherence (Hardmeier et al., 2012), syntactic features (Post, 2011), word posterior probability (Ueffing and Ney, 2007), etc.

- identify correct phrases to protect them from rewriting

- adapt rewriter’s objective function to the sentence

- use a paraphrase operation rewriting the source sentence to produce new target phrases (Marie and Max, 2013)

- use automatic alternative reference translations (Madnani and Dorr, 2013)

- use rewriter in interaction with human translators
Future work

- exploit more features: lexical-coherence (Hardmeier et al., 2012), syntactic features (Post, 2011), word posterior probability (Ueffing and Ney, 2007), etc.
- identify correct phrases to protect them from rewriting
- adapt rewriter’s objective function to the sentence
- use a paraphrase operation rewriting the source sentence to produce new target phrases (Marie and Max, 2013)
- use automatic alternative reference translations (Madnani and Dorr, 2013)
- use rewriter in interaction with human translators
Confidence-based Rewriting of Machine Translation Output

Benjamin Marie & Aurélien Max

emnlp 2014


