Reordering Problem and Solutions

The Word

Alok Parlikar

Language Technologies Institute
Carnegie Mellon University

March 19, 2008 / Advanced MT Seminar
Introduction

POS-based Reordering

Syntactic Reordering

Conclusion

What is going on?

Hey, Welcome to my neighborhood. Where do you come from?

Hi. Thanks. Just a few blocks away, actually... They are asking a lot of people to exchange places... What is going on? 😊

Another evaluation? Someone might be trying source-reordering experiments!

Don’t they know about GIGO?

I think they do. They must only be figuring out which ‘garbage in’ minimizes ‘garbage out’.

Man

Woman
Ok Ok..

Aaah! OK!
I will try to speak normal English now.

I wish I were Shakespeare....
Outline

1. Introduction
   - Reordering in Phrase-based SMT
   - Word Orders between Language Pairs

2. POS-based Reordering
   - Popović, Ney (2006)
   - Crego, Mariño (2006)
   - Rottmann, Vogel (2007)

3. Syntactic Reordering
   - Nguyen, Shimazu (2006)
   - Li et al. (2007)

4. Conclusion
Outline

1 Introduction
   - Reordering in Phrase-based SMT
     - Word Orders between Language Pairs

2 POS-based Reordering
   - Popović, Ney (2006)
   - Crego, Mariño (2006)
   - Rottmann, Vogel (2007)

3 Syntactic Reordering
   - Nguyen, Shimazu (2006)
   - Li et al. (2007)

4 Conclusion
Reordering in Phrase-based SMT

- Phrase based SMT models has achieved the state-of-the-art performance. These models have several advantages, such as word choice, idomatic expression recognition, and local restructuring.
- There still are potential limitations when it comes to modelling word-order differences between languages.
- Use of reordering allows for important improvement in translation accuracy.
- Arbitrary word reorderings could be permitted...
- Typically used reordering: Distance based reordering.
- Are somehow ‘non-linguistic’
Introducing Linguistic Information

Source Sentence Reordering

Transform the source sentence so that the order of words conforms to that in the target language.

- How to model the word reordering from source to target?
- How to score different reorderings?
- How to apply the model at run-time?
Introducing Linguistic Information

Source Sentence Reordering

Transform the source sentence so that the order of words conforms to that in the target language.

- How to model the word reordering from source to target?
- How to score different reorderings?
- How to apply the model at run-time?
Introducing Linguistic Information

Source Sentence Reordering

Transform the source sentence so that the order of words conforms to that in the target language.

- How to model the word reordering from source to target?
- How to score different reorderings?
- How to apply the model at run-time?
Introducing Linguistic Information

Source Sentence Reordering
Transform the source sentence so that the order of words conforms to that in the target language.

- How to model the word reordering from source to target?
- How to score different reorderings?
- How to apply the model at run-time?
Outline

1. Introduction
   - Reordering in Phrase-based SMT
   - Word Orders between Language Pairs

2. POS-based Reordering
   - Popović, Ney (2006)
   - Crego, Mariño (2006)
   - Rottmann, Vogel (2007)

3. Syntactic Reordering
   - Nguyen, Shimazu (2006)
   - Li et al. (2007)

4. Conclusion
Reordering phenomena between languages

- Most papers report results on European languages
- French, Spanish, German, English
- Also: Asian languages (Vietnamese)
French, Spanish ↔ English

- Local Reorderings
- Most adjectives come after the noun in French, Spanish. In English, Adjectives come before the nouns.

N  ADJ ⇔ ADJ  N

Example

- French: train *rouge*
- English: *red* train
German ↔ English

- Global Reorderings
- Infinitives and Past Participles are placed at the end of a clause in German. In English, they usually occur towards the beginning of the clause.
- Detached verb prefixes also go to the end of the clause.

Example
- German: Ich werde morgen nachmittag ... ankomen
- English: I will arrive tomorrow afternoon ...
Vietnamese ↔ English

- SVO word order, similar to English
- WH-movement is significantly different. (The interrogative word is not moved to the beginning of the sentence).
- Most yes-no questions end in an interrogative word.
- Most phrases are head-final.

**Example**

- Vietnamese: BOOK ’s FRIEND HIS
- English: his friend’s book
Outline

1. Introduction
   - Reordering in Phrase-based SMT
   - Word Orders between Language Pairs

2. POS-based Reordering
   - Popović, Ney (2006)
   - Crego, Mariño (2006)
   - Rottmann, Vogel (2007)

3. Syntactic Reordering
   - Nguyen, Shimazu (2006)
   - Li et al. (2007)

4. Conclusion
POS-based Word Reordering Models for SMT
Popović, Ney (2006)

- Languages: English, Spanish, German
- Based entirely on POS. Additional syntactic tools (parsers) not required.
- Limited range of reordering phenomena:
  1. Adjective-Noun reordering in Spanish
     - From Spanish to English/German: Move adjective before noun group
     - From English/German to Spanish: Move adjective after noun group
  2. Verb reordering in German
     - From Spanish/English to German: Move infinitive or past participle to end of the clause. Keep auxiliary verb in original position.
Experimental Setup
Popović, Ney (2006)

- Europarl Corpus: 700K sentences
- POS Taggers: FreeLing, ENGCG, GERCG
- Trilingual Corpus: 670K sentences
- Studied effect of data sparsity, and training-corpus reordering.
- RWTH SMT System used for decoding.

<table>
<thead>
<tr>
<th></th>
<th>Spanish</th>
<th>English</th>
<th>German</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Train:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentences</td>
<td>730740</td>
<td>751088</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>15724914</td>
<td>15222146</td>
<td>15257678</td>
<td>16052330</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>113882</td>
<td>72739</td>
<td>205374</td>
<td>74708</td>
</tr>
<tr>
<td>Singletons [%]</td>
<td>39.2</td>
<td>38.3</td>
<td>49.8</td>
<td>38.3</td>
</tr>
<tr>
<td><strong>Dev:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentences</td>
<td>2000</td>
<td>2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>60628</td>
<td>58655</td>
<td>55147</td>
<td>58655</td>
</tr>
<tr>
<td>Distinct Words</td>
<td>8182</td>
<td>6547</td>
<td>9213</td>
<td>6547</td>
</tr>
<tr>
<td>OOVs [%]</td>
<td>0.4</td>
<td>0.2</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td><strong>Test:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentences</td>
<td>2000</td>
<td>2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>60332</td>
<td>57951</td>
<td>54260</td>
<td>57951</td>
</tr>
<tr>
<td>Distinct Words</td>
<td>8279</td>
<td>6496</td>
<td>9048</td>
<td>6496</td>
</tr>
<tr>
<td>OOVs [%]</td>
<td>0.4</td>
<td>0.2</td>
<td>0.7</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Experimental Setup

Popović, Ney (2006)

- Europarl Corpus: 700K sentences
- POS Taggers: FreeLing, ENGCG, GERCG
- Trilingual Corpus: 670K sentences
- Studied effect of data sparsity, and training-corpus reordering.
- RWTH SMT System used for decoding.

<table>
<thead>
<tr>
<th></th>
<th>Spanish</th>
<th>English</th>
<th>German</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train: Sentences</td>
<td>730740</td>
<td>751088</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>15724914</td>
<td>15222146</td>
<td>15257678</td>
<td>16052330</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>113882</td>
<td>72739</td>
<td>205374</td>
<td>74708</td>
</tr>
<tr>
<td>Singletons [%]</td>
<td>39.2</td>
<td>38.3</td>
<td>49.8</td>
<td>38.3</td>
</tr>
<tr>
<td>Dev: Sentences</td>
<td>2000</td>
<td>2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>60628</td>
<td>58655</td>
<td>55147</td>
<td>58655</td>
</tr>
<tr>
<td>Distinct Words</td>
<td>8182</td>
<td>6547</td>
<td>9213</td>
<td>6547</td>
</tr>
<tr>
<td>OOVs [%]</td>
<td>0.4</td>
<td>0.2</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>Test: Sentences</td>
<td>2000</td>
<td>2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>60332</td>
<td>57951</td>
<td>54260</td>
<td>57951</td>
</tr>
<tr>
<td>Distinct Words</td>
<td>8279</td>
<td>6496</td>
<td>9048</td>
<td>6496</td>
</tr>
<tr>
<td>OOVs [%]</td>
<td>0.4</td>
<td>0.2</td>
<td>0.7</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Experimental Setup
Popović, Ney (2006)

- Europarl Corpus: 700K sentences
- POS Taggers: FreeLing, ENGCG, GERCG
- Trilingual Corpus: 670K sentences
- Studied effect of data sparsity, and training-corpus reordering.
- RWTH SMT System used for decoding.

<table>
<thead>
<tr>
<th></th>
<th>Spanish</th>
<th>English</th>
<th>German</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train: Sentences</td>
<td>730740</td>
<td>751088</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>15724914</td>
<td>15222146</td>
<td>15257678</td>
<td>16052330</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>113882</td>
<td>72739</td>
<td>205374</td>
<td>74708</td>
</tr>
<tr>
<td>Singletons [%]</td>
<td>39.2</td>
<td>38.3</td>
<td>49.8</td>
<td>38.3</td>
</tr>
<tr>
<td>Dev: Sentences</td>
<td>2000</td>
<td>2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>60628</td>
<td>58655</td>
<td>55147</td>
<td>58655</td>
</tr>
<tr>
<td>Distinct Words</td>
<td>8182</td>
<td>6547</td>
<td>9213</td>
<td>6547</td>
</tr>
<tr>
<td>OOVs [%]</td>
<td>0.4</td>
<td>0.2</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>Test: Sentences</td>
<td>2000</td>
<td>2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>60332</td>
<td>57951</td>
<td>54260</td>
<td>57951</td>
</tr>
<tr>
<td>Distinct Words</td>
<td>8279</td>
<td>6496</td>
<td>9048</td>
<td>6496</td>
</tr>
<tr>
<td>OOVs [%]</td>
<td>0.4</td>
<td>0.2</td>
<td>0.7</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Experimental Setup
Popović, Ney (2006)

- Europarl Corpus: 700K sentences
- POS Taggers: FreeLing, ENGCG, GERCG
- Trilingual Corpus: 670K sentences
- Studied effect of data sparsity, and training-corpus reordering.

- RWTH SMT System used for decoding.

<table>
<thead>
<tr>
<th></th>
<th>Spanish</th>
<th>English</th>
<th>German</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentences</td>
<td>730740</td>
<td>751088</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>15724914</td>
<td>15222146</td>
<td>15257678</td>
<td>16052330</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>113882</td>
<td>72739</td>
<td>205374</td>
<td>74708</td>
</tr>
<tr>
<td>Singletons [%]</td>
<td>39.2</td>
<td>38.3</td>
<td>49.8</td>
<td>38.3</td>
</tr>
<tr>
<td>Dev</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentences</td>
<td>2000</td>
<td>2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>60628</td>
<td>58655</td>
<td>55147</td>
<td>58655</td>
</tr>
<tr>
<td>Distinct Words</td>
<td>8182</td>
<td>6547</td>
<td>9213</td>
<td>6547</td>
</tr>
<tr>
<td>OOVs [%]</td>
<td>0.4</td>
<td>0.2</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>Test</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentences</td>
<td>2000</td>
<td>2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>60332</td>
<td>57951</td>
<td>54260</td>
<td>57951</td>
</tr>
<tr>
<td>Distinct Words</td>
<td>8279</td>
<td>6496</td>
<td>9048</td>
<td>6496</td>
</tr>
<tr>
<td>OOVs [%]</td>
<td>0.4</td>
<td>0.2</td>
<td>0.7</td>
<td>0.2</td>
</tr>
</tbody>
</table>
**Experimental Setup**

Popović, Ney (2006)

- Europarl Corpus: 700K sentences
- POS Taggers: FreeLing, ENGCG, GERCG
- Trilingual Corpus: 670K sentences
- Studied effect of data sparsity, and training-corpus reordering.
- RWTH SMT System used for decoding.

<table>
<thead>
<tr>
<th></th>
<th>Spanish</th>
<th>English</th>
<th>German</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Train:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentences</td>
<td>730740</td>
<td>751088</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>15724914</td>
<td>15222146</td>
<td>15257678</td>
<td>16052330</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>113882</td>
<td>72739</td>
<td>205374</td>
<td>74708</td>
</tr>
<tr>
<td>Singletons [%]</td>
<td>39.2</td>
<td>38.3</td>
<td>49.8</td>
<td>38.3</td>
</tr>
<tr>
<td><strong>Dev:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentences</td>
<td>2000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>60628</td>
<td>58655</td>
<td>55147</td>
<td>58655</td>
</tr>
<tr>
<td>Distinct Words</td>
<td>8182</td>
<td>6547</td>
<td>9213</td>
<td>6547</td>
</tr>
<tr>
<td>OOVs [%]</td>
<td>0.4</td>
<td>0.2</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td><strong>Test:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentences</td>
<td>2000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Words+Punctuation</td>
<td>60332</td>
<td>57951</td>
<td>54260</td>
<td>57951</td>
</tr>
<tr>
<td>Distinct Words</td>
<td>8279</td>
<td>6496</td>
<td>9048</td>
<td>6496</td>
</tr>
<tr>
<td>OOVs [%]</td>
<td>0.4</td>
<td>0.2</td>
<td>0.7</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Experimental Results
Popović, Ney (2006)

- Spanish To English
- English To Spanish
- English To German
- Spanish To German

<table>
<thead>
<tr>
<th>Spanish→English</th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WER</td>
<td>PER</td>
</tr>
<tr>
<td>700k reordered</td>
<td>56.7</td>
<td>42.8</td>
</tr>
<tr>
<td>rest reorder adjective</td>
<td>56.3</td>
<td>42.2</td>
</tr>
<tr>
<td></td>
<td>57.1</td>
<td>45.3</td>
</tr>
<tr>
<td>rest reorder adjective</td>
<td>56.6</td>
<td>44.9</td>
</tr>
<tr>
<td>7k reordered</td>
<td>65.9</td>
<td>49.4</td>
</tr>
<tr>
<td>rest reorder adjective</td>
<td>64.2</td>
<td>48.3</td>
</tr>
<tr>
<td></td>
<td>64.6</td>
<td>51.6</td>
</tr>
<tr>
<td>rest reorder adjective</td>
<td>64.4</td>
<td>51.0</td>
</tr>
</tbody>
</table>
## Experimental Results

Popović, Ney (2006)

- Spanish To English
- **English To Spanish**
- English To German
- Spanish To German

<table>
<thead>
<tr>
<th>English→Spanish</th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WER</td>
<td>PER</td>
</tr>
<tr>
<td>700k reordered</td>
<td>59.0</td>
<td>45.4</td>
</tr>
<tr>
<td></td>
<td>58.6</td>
<td>45.2</td>
</tr>
<tr>
<td>reordered</td>
<td>baseline reorder adjectives</td>
<td>57.0</td>
</tr>
<tr>
<td></td>
<td>56.8</td>
<td>46.4</td>
</tr>
<tr>
<td>rest</td>
<td>baseline reorder adjectives</td>
<td>67.6</td>
</tr>
<tr>
<td></td>
<td>66.7</td>
<td>51.6</td>
</tr>
<tr>
<td>7k reordered</td>
<td>baseline reorder adjectives</td>
<td>65.3</td>
</tr>
<tr>
<td></td>
<td>65.2</td>
<td>52.8</td>
</tr>
</tbody>
</table>
Experimental Results
Popović, Ney (2006)

- Spanish To English
- English To Spanish
- **English To German**
- Spanish To German

<table>
<thead>
<tr>
<th>English→German</th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WER</td>
<td>PER</td>
</tr>
<tr>
<td>700k reordered</td>
<td>70.3</td>
<td>57.8</td>
</tr>
<tr>
<td>reordered verbs</td>
<td>69.7</td>
<td>56.9</td>
</tr>
<tr>
<td>rest reordered verbs</td>
<td>64.9</td>
<td>54.0</td>
</tr>
<tr>
<td>rest reordered verbs</td>
<td>64.6</td>
<td>53.8</td>
</tr>
<tr>
<td>7k reordered</td>
<td>79.3</td>
<td>62.5</td>
</tr>
<tr>
<td>reordered verbs</td>
<td>78.9</td>
<td>62.4</td>
</tr>
<tr>
<td>rest reordered verbs</td>
<td>73.8</td>
<td>60.6</td>
</tr>
<tr>
<td>rest reordered verbs</td>
<td>74.0</td>
<td>60.6</td>
</tr>
</tbody>
</table>
### Experimental Results

Popović, Ney (2006)

- Spanish To English
- English To Spanish
- English To German
- Spanish To German

<table>
<thead>
<tr>
<th>Spanish→German</th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WER</td>
<td>PER</td>
</tr>
<tr>
<td>baseline</td>
<td>68.4</td>
<td>55.0</td>
</tr>
<tr>
<td>reorder adjectives</td>
<td>68.2</td>
<td>55.0</td>
</tr>
<tr>
<td>reorder verbs</td>
<td>68.0</td>
<td>55.0</td>
</tr>
<tr>
<td>reorder adjectives + verbs</td>
<td>67.9</td>
<td>54.7</td>
</tr>
<tr>
<td>6k</td>
<td></td>
<td></td>
</tr>
<tr>
<td>baseline</td>
<td>78.9</td>
<td>63.3</td>
</tr>
<tr>
<td>reorder adjectives</td>
<td>78.3</td>
<td>62.9</td>
</tr>
<tr>
<td>reorder verbs</td>
<td>78.8</td>
<td>63.3</td>
</tr>
<tr>
<td>reorder adjectives + verbs</td>
<td>78.5</td>
<td>63.0</td>
</tr>
</tbody>
</table>
Introduction

Reordering in Phrase-based SMT
Word Orders between Language Pairs

POS-based Reordering

Popović, Ney (2006)
Crego, Mariño (2006)
Rottmann, Vogel (2007)

Syntactic Reordering

Nguyen, Shimazu (2006)
Li et al. (2007)

Conclusion
Integration of POS-based source reordering into SMT
Crego, Mariño (2006)

- Languages: English to Spanish, and Spanish to English
- Corpus used: Europarl (1.28 M Sentences)
- POS Taggers Used: TNT (English), FreeLing(Spanish)
- Extract reordering patterns from corpus
- Build ‘extended search graph’ by applying reordering patterns to source sentence
- Use MARIE decoder (n-gram based SMT)
Reordering Framework
Crego, Mariño (2006)

- Get bi-directional GIZA alignments for the corpus, and take the UNION.
- Identify all crossing alignments produced.
- For each crossing:
  - Take the sequence of source side tags between the crossings
  - Create a rewrite pattern based on the order in which the source tags appear on the target side.
Reordering Framework
Crego, Mariño (2006)

- Get bi-directional GIZA alignments for the corpus, and take the UNION.
- Identify all crossing alignments produced.
- For each crossing:
  - Take the sequence of source side tags between the crossings
  - Create a rewrite pattern based on the order in which the source tags appear on the target side.
Get bi-directional GIZA alignments for the corpus, and take the UNION.

Identify all crossing alignments produced.

For each crossing:

- Take the sequence of source side tags between the crossings
- Create a rewrite pattern based on the order in which the source tags appear on the target side.
Reordering Framework
Crego, Mariño (2006)

- Get bi-directional GIZA alignments for the corpus, and take the UNION.
- Identify all crossing alignments produced.
- For each crossing:
  - Take the sequence of source side tags between the crossings
  - Create a rewrite pattern based on the order in which the source tags appear on the target side.
Reordering Framework
Crego, Mariño (2006)

- Get bi-directional GIZA alignments for the corpus, and take the UNION.
- Identify all crossing alignments produced.
- For each crossing:
  - Take the sequence of source side tags between the crossings
  - Create a rewrite pattern based on the order in which the source tags appear on the target side.
Reordering Framework
Crego, Mariño (2006)

- For each crossing:
  - Take the sequence of source side tags between the crossings
  - Create a rewrite pattern based on the order in which the source tags appear on the target side.

I ideas excelentes y constructivas

excellent and constructive ideas
Reordering Framework
Crego, Mariño (2006)

For each crossing:
- Take the sequence of source side tags between the crossings
- Create a rewrite pattern based on the order in which the source tags appear on the target side.

Idea excelente y constructivas
excellent and constructive ideas
For each crossing:
- Take the sequence of source side tags between the crossings
- Create a rewrite pattern based on the order in which the source tags appear on the target side.
For each crossing:

- Take the sequence of source side tags between the crossings
- Create a rewrite pattern based on the order in which the source tags appear on the target side.
Reordering Framework
Crego, Mariño (2006)

- For each crossing:
  - Take the sequence of source side tags between the crossings
  - Create a rewrite pattern based on the order in which the source tags appear on the target side.

Idea excelentes y constructivas
excellent and constructive ideas
Reordering Framework
Crego, Mariño (2006)

- For each crossing:
  - Take the sequence of source side tags between the crossings
  - Create a rewrite pattern based on the order in which the source tags appear on the target side.
Rule Filtering
Crego, Mariño (2006)

- Large number of rules can be extracted from the corpus
- Most of the rules appear due to wrong word alignments!
- Apply the following filters:
  - Source and Target Phrases (where crossing occurs) must be atmost 4 words different in length.
  - Maximum length of a rewrite pattern is 8.
  - A pattern must occur at least 1000 times.
  - \[
    \frac{n(\text{pattern})}{n(\text{sourcewords})} > 0.2
  \]
- Rules left after filtering: 29.
- Errors still remain!
Rule Filtering
Crego, Mariño (2006)

- Large number of rules can be extracted from the corpus
- Most of the rules appear due to wrong word alignments!
- Apply the following filters:
  1. Source and Target Phrases (where crossing occurs) must be at most 4 words different in length.
  2. Maximum length of a rewrite pattern is 8.
  3. A pattern must occur at least 1000 times.
  4. \[ \frac{n(\text{pattern})}{n(\text{source words})} > 0.2 \]
- Rules left after filtering: 29.
- Errors still remain!
Large number of rules can be extracted from the corpus
Most of the rules appear due to wrong word alignments!
Apply the following filters:
1. Source and Target Phrases (where crossing occurs) must be at most 4 words different in length.
2. Maximum length of a rewrite pattern is 8.
3. A pattern must occur at least 1000 times.
4. \[ \frac{n(pattern)}{n(sourcewords)} > 0.2 \]
Rules left after filtering: 29.
Errors still remain!
Large number of rules can be extracted from the corpus
Most of the rules appear due to wrong word alignments!
Apply the following filters:

1. Source and Target Phrases (where crossing occurs) must be atmost 4 words different in length.
2. Maximum length of a rewrite pattern is 8.
3. A pattern must occur at least 1000 times.
4. \[
\frac{n(\text{pattern})}{n(\text{sourcewords})} > 0.2
\]

Rules left after filtering: 29.
Errors still remain!
Rule Filtering
Crego, Mariño (2006)

- Large number of rules can be extracted from the corpus
- Most of the rules appear due to wrong word alignments!
- Apply the following filters:
  1. Source and Target Phrases (where crossing occurs) must be atmost 4 words different in length.
  2. Maximum length of a rewrite pattern is 8.
  3. A pattern must occur at least 1000 times.
  4. \( \frac{n(\text{pattern})}{n(\text{sourcewords})} > 0.2 \)

- Rules left after filtering: 29.
- Errors still remain!
Large number of rules can be extracted from the corpus
Most of the rules appear due to wrong word alignments!

Apply the following filters:

1. Source and Target Phrases (where crossing occurs) must be atmost 4 words different in length.
2. Maximum length of a rewrite pattern is 8.
3. A pattern must occur at least 1000 times.
4. \( \frac{n(\text{pattern})}{n(\text{source words})} > 0.2 \)

Rules left after filtering: 29.
Errors still remain!
Large number of rules can be extracted from the corpus
Most of the rules appear due to wrong word alignments!
Apply the following filters:
1. Source and Target Phrases (where crossing occurs) must be almost 4 words different in length.
2. Maximum length of a rewrite pattern is 8.
3. A pattern must occur at least 1000 times.
4. \( \frac{n(pattern)}{n(sourcewords)} > 0.2 \)

Rules left after filtering: 29.
Errors still remain!
Large number of rules can be extracted from the corpus
Most of the rules appear due to wrong word alignments!
Apply the following filters:
1. Source and Target Phrases (where crossing occurs) must be atmost 4 words different in length.
2. Maximum length of a rewrite pattern is 8.
3. A pattern must occur at least 1000 times.
4. \( \frac{n(pattern)}{n(source words)} > 0.2 \)

Rules left after filtering: 29.

Errors still remain!
Rule Filtering
Crego, Mariño (2006)

- Large number of rules can be extracted from the corpus
- Most of the rules appear due to wrong word alignments!
- Apply the following filters:
  1. Source and Target Phrases (where crossing occurs) must be at most 4 words different in length.
  2. Maximum length of a rewrite pattern is 8.
  3. A pattern must occur at least 1000 times.
  4. $\frac{n(pattern)}{n(sourcewords)} > 0.2$
- Rules left after filtering: 29.
- Errors *still* remain!
Building an Extended Search Graph
Crego, Mariño (2006)

- Take the POS tagged input sentence
- Consider all applicable reordering rules
- Build a monotone search path for the input
- Apply each rule, and add entry to the path.
Building an Extended Search Graph
Crego, Mariño (2006)

- Take the POS tagged input sentence
- Consider all applicable reordering rules
- Build a monotone search path for the input
- Apply each rule, and add entry to the path.

*programa ambicioso y realista*

```
NC  AQ   CC  AQ
NC  AQ
NC  AQ  CC  -1  2  0
NC  AQ  CC  AQ  -1  2  3  0
```
Building an Extended Search Graph
Crego, Mariño (2006)

- Take the POS tagged input sentence
- Consider all applicable reordering rules
- Build a monotone search path for the input
- Apply each rule, and add entry to the path.
Building an Extended Search Graph
Crego, Mariño (2006)

- Take the POS tagged input sentence
- Consider all applicable reordering rules
- Build a monotone search path for the input
- Apply each rule, and add entry to the path.
Building an Extended Search Graph
Crego, Mariño (2006)

- Take the POS tagged input sentence
- Consider all applicable reordering rules
- Build a monotone search path for the input
- Apply each rule, and add entry to the path.
Building an Extended Search Graph
Crego, Mariño (2006)

- Take the POS tagged input sentence
- Consider all applicable reordering rules
- Build a monotone search path for the input
- Apply each rule, and add entry to the path.
Source side training corpus was reordered using the given rewrite patterns, and a 5-gram source-side LM was used.

If more than one pattern can be applied, priority goes to the longest pattern.

Three comparable systems:

1. baseline: Monotone Search
2. rgraph: Monotone search within reorder graphs
3. pos: Monotone search within reorder graphs, with source side LM

Two reference translations per sentence
# Results

Crego, Mariño (2006)

<table>
<thead>
<tr>
<th>Conf</th>
<th>bleu'</th>
<th>bleu</th>
<th>nist</th>
<th>mwer</th>
<th>per</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spanish-to-English</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>base</td>
<td>.529</td>
<td>.552</td>
<td>10.69</td>
<td>34.40</td>
<td>25.32</td>
</tr>
<tr>
<td>rgraph</td>
<td>.533</td>
<td>.556</td>
<td>10.70</td>
<td>34.23</td>
<td>25.50</td>
</tr>
<tr>
<td>pos</td>
<td>.539</td>
<td>.564</td>
<td>10.75</td>
<td>33.75</td>
<td>25.41</td>
</tr>
<tr>
<td><strong>English-to-Spanish</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>base</td>
<td>.481</td>
<td>.480</td>
<td>9.84</td>
<td>41.18</td>
<td>31.11</td>
</tr>
<tr>
<td>rgraph</td>
<td>.490</td>
<td>.485</td>
<td>9.81</td>
<td>41.15</td>
<td>31.87</td>
</tr>
<tr>
<td>pos</td>
<td>.491</td>
<td>.489</td>
<td>9.91</td>
<td>40.29</td>
<td>31.27</td>
</tr>
</tbody>
</table>
## Results

Crego, Mariño (2006)

<table>
<thead>
<tr>
<th>Pattern</th>
<th>train</th>
<th>dev</th>
<th>test</th>
<th>swap</th>
<th>error</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC RG AQ CC AQ → 1 2 3 4 0</td>
<td>1,406</td>
<td>123</td>
<td>170</td>
<td>2</td>
<td>0</td>
<td>ideas muy sencillas y elementales</td>
</tr>
<tr>
<td>NC AQ CC AQ → 1 2 3 0</td>
<td>27,119</td>
<td>132</td>
<td>17</td>
<td>2</td>
<td>0</td>
<td>programa ambicioso y realista</td>
</tr>
<tr>
<td>NC AQ RG AQ → 2 3 1 0</td>
<td>1,971</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>control fronterizo más estricto</td>
</tr>
<tr>
<td>NC CC NC AQ → 3 0 1 2</td>
<td>3,355</td>
<td>6</td>
<td>12</td>
<td>6</td>
<td>3</td>
<td>mezquitas y centros islámicos</td>
</tr>
<tr>
<td>NC RG AQ CC → 1 2 3 0</td>
<td>2,226</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>ideas muy sencillas y</td>
</tr>
<tr>
<td>AQ RG AQ → 1 2 0</td>
<td>2,777</td>
<td>21</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>europea más sólida</td>
</tr>
<tr>
<td>NC AQ AQ → 2 1 0</td>
<td>35,661</td>
<td>11</td>
<td>24</td>
<td>18</td>
<td>3</td>
<td>decisiones políticas delicadas</td>
</tr>
<tr>
<td>NC RG AQ → 1 2 0</td>
<td>32,887</td>
<td>0</td>
<td>35</td>
<td>26</td>
<td>1</td>
<td>ideas muy sencillas</td>
</tr>
<tr>
<td>NC RG RG → 1 2 0</td>
<td>1,473</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>texto mucho más</td>
</tr>
<tr>
<td>NC AQ → 1 0</td>
<td>877,580</td>
<td>113</td>
<td>142</td>
<td>110</td>
<td>16</td>
<td>preguntas serias</td>
</tr>
<tr>
<td>NC RG → 1 0</td>
<td>54,968</td>
<td>27</td>
<td>47</td>
<td>7</td>
<td>7</td>
<td>actividades aparentemente</td>
</tr>
<tr>
<td>AQ AQ → 1 0</td>
<td>46,509</td>
<td>14</td>
<td>40</td>
<td>4</td>
<td>2</td>
<td>medioambientales europeas</td>
</tr>
<tr>
<td>RN VM → 1 0</td>
<td>45,777</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>no promuevan</td>
</tr>
<tr>
<td>RG VA → 1 0</td>
<td>9,824</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>ahora habíamos</td>
</tr>
<tr>
<td>AQ RG → 1 0</td>
<td>8,701</td>
<td>11</td>
<td>21</td>
<td>4</td>
<td>2</td>
<td>suficiente todavía</td>
</tr>
<tr>
<td>RG VS → 1 0</td>
<td>5,043</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>supuestamente somos</td>
</tr>
<tr>
<td>VM PP → 1 0</td>
<td>4,769</td>
<td>6</td>
<td>13</td>
<td>12</td>
<td>2</td>
<td>estar ustedes</td>
</tr>
</tbody>
</table>

| Total (17) | 1,162,046 | 231 | 379 | 214 | 42 |
Outline

1. Introduction
   - Reordering in Phrase-based SMT
   - Word Orders between Language Pairs

2. POS-based Reordering
   - Popović, Ney (2006)
   - Crego, Mariño (2006)
   - Rottmann, Vogel (2007)

3. Syntactic Reordering
   - Nguyen, Shimazu (2006)
   - Li et al. (2007)

4. Conclusion
Word Reordering in SMT with POS based DM
Rottmann, Vogel (2007)

- Approach similar to Crego-Mariño
- POS based rules for reordering source text
- Use lattice to represent reorderings, and keep decoding monotone
- Use context information to help differentiate reorderings that are purely context based.
Learning Rules
Rottmann, Vogel (2007)

- Get alignments for bilingual corpus. Use POS tagger to get source side tags.
- Find crossings in alignment. Extract a reordering rule for every crossing.
- A rule which is observed as a part of a longer reordering is stored only if it also occurs as the longest reordering sequence in some other sentence pair.
- Filter rules for 5 or more occurrences. Assign rule scores using relative frequency.

Example:
Learning Rules
Rottmann, Vogel (2007)

- Get alignments for bilingual corpus. Use POS tagger to get source side tags.
- Find crossings in alignment. Extract a reordering rule for every crossing.
- A rule which is observed as a part of a longer reordering is stored only if it also occurs as the longest reordering sequence in some other sentence pair.
- Filter rules for 5 or more occurrences. Assign rule scores using relative frequency.

Example:
Learning Rules
Rottmann, Vogel (2007)

- Get alignments for bilingual corpus. Use POS tagger to get source side tags.
- Find crossings in alignment. Extract a reordering rule for every crossing.
- A rule which is observed as a part of a longer reordering is stored only if it also occurs as the longest reordering sequence in some other sentence pair.
- Filter rules for 5 or more occurrences. Assign rule scores using relative frequency.

Example:
Learning Rules
Rottmann, Vogel (2007)

- Get alignments for bilingual corpus. Use POS tagger to get source side tags.
- Find crossings in alignment. Extract a reordering rule for every crossing.
- A rule which is observed as a part of a longer reordering is stored only if it also occurs as the longest reordering sequence in some other sentence pair.
- Filter rules for 5 or more occurrences. Assign rule scores using relative frequency.

Example:
Learning Rules
Rottmann, Vogel(2007)

- Get alignments for bilingual corpus. Use POS tagger to get source side tags.
- Find crossings in alignment. Extract a reordering rule for every crossing.
- A rule which is observed as a part of a longer reordering is stored only if it also occurs as the longest reordering sequence in some other sentence pair.
- Filter rules for 5 or more occurrences. Assign rule scores using relative frequency.

<table>
<thead>
<tr>
<th>source sequence</th>
<th>rule</th>
<th>freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDAT NN VVINF</td>
<td>3 1 2</td>
<td>0.60</td>
</tr>
<tr>
<td>VAFIN :: PDAT NN VVINF</td>
<td>3 1 2</td>
<td>0.63</td>
</tr>
<tr>
<td>KOUI :: PDAT NN VVINF</td>
<td>3 2 2</td>
<td>0.88</td>
</tr>
<tr>
<td>moechte :: PDAT NN VVINF</td>
<td>3 1 2</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Example:
Rottmann, Vogel (2007)

- Start with the POS tags of the input sentence.
- Match the POS tags to the rules and expand the lattice to reflect new word orders. Use context information if applicable.
- Once the lattice is built, assign rule scores.
Applying Rules
Rottmann, Vogel (2007)

- Start with the POS tags of the input sentence.
- Match the POS tags to the rules and expand the lattice to reflect new word orders. Use context information if applicable.
- Once the lattice is built, assign rule scores.
Applying Rules
Rottmann, Vogel(2007)

- Start with the POS tags of the input sentence.
- Match the POS tags to the rules and expand the lattice to reflect new word orders. Use context information if applicable.
- Once the lattice is built, assign rule scores.
Experimental Setup
Rottmann, Vogel (2007)

- Europarl corpus used. Two references for English and Spanish. One reference for German-English.
- POS Taggers used: Brill (Eng), Stuttgart Tree Tagger (German).
- Rules of up to length 15 extracted.
Experimental Setup
Rottmann, Vogel (2007)

- Europarl corpus used. Two references for English and Spanish. One reference for German-English.
- POS Taggers used: Brill (Eng), Stuttgart Tree Tagger (German).
- Rules of upto length 15 extracted.
Experimental Setup
Rottmann, Vogel(2007)

- Europarl corpus used. Two references for English and Spanish. One reference for German-English.
- POS Taggers used: Brill (Eng), Stuttgart Tree Tagger (German).
- Rules of upto length 15 extracted.
## Results

Rottmann, Vogel (2007)

<table>
<thead>
<tr>
<th>System Context</th>
<th>Threshold</th>
<th># en \rightarrow es</th>
<th># en \rightarrow de</th>
<th># de \rightarrow en</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Rules Learned</td>
<td>Rule Matches</td>
<td>Rules Learned</td>
</tr>
<tr>
<td>no</td>
<td>0.05</td>
<td>21388</td>
<td>12715</td>
<td>7929</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>6848</td>
<td>7740</td>
<td>4061</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>2321</td>
<td>4247</td>
<td>1291</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>1136</td>
<td>3369</td>
<td>469</td>
</tr>
<tr>
<td>yes</td>
<td>0.01</td>
<td>72772</td>
<td>21119</td>
<td>32380</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>46014</td>
<td>6888</td>
<td>22836</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>25962</td>
<td>4924</td>
<td>15941</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>15304</td>
<td>3461</td>
<td>8462</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>en \rightarrow es</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (RO3)</td>
<td>49.98</td>
</tr>
<tr>
<td>POS no Context 0.05</td>
<td>50.36</td>
</tr>
<tr>
<td>POS no Context 0.1</td>
<td>51.09</td>
</tr>
<tr>
<td>POS no Context 0.2</td>
<td>50.66</td>
</tr>
<tr>
<td>POS no Context 0.3</td>
<td>50.59</td>
</tr>
<tr>
<td>POS + Context 0.01</td>
<td>50.92</td>
</tr>
<tr>
<td>POS + Context 0.05</td>
<td>50.90</td>
</tr>
<tr>
<td>POS + Context 0.1</td>
<td>50.84</td>
</tr>
<tr>
<td>POS + Context 0.2</td>
<td>50.74</td>
</tr>
<tr>
<td>unseen Baseline (RO3)</td>
<td>48.51</td>
</tr>
<tr>
<td>unseen no Context</td>
<td>49.57</td>
</tr>
<tr>
<td>unseen with Context</td>
<td>49.49</td>
</tr>
</tbody>
</table>
## Results

Rottmann, Vogel (2007)

<table>
<thead>
<tr>
<th>System Context</th>
<th>Threshold</th>
<th># en → es</th>
<th># en → de</th>
<th># de → en</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Rules</td>
<td>Rules</td>
<td>Rules</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Learned</td>
<td>Matches</td>
<td>Learned</td>
</tr>
<tr>
<td>no</td>
<td>0.05</td>
<td>21388</td>
<td>12715</td>
<td>7929</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>6848</td>
<td>7740</td>
<td>4061</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>2321</td>
<td>4247</td>
<td>1291</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>1136</td>
<td>3369</td>
<td>469</td>
</tr>
<tr>
<td>yes</td>
<td>0.01</td>
<td>72772</td>
<td>21119</td>
<td>32380</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>46014</td>
<td>6888</td>
<td>22836</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>25962</td>
<td>4924</td>
<td>15941</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>15304</td>
<td>3461</td>
<td>8462</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>en → de</th>
<th>de → en</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (RO3)</td>
<td>18.92</td>
<td>25.64</td>
</tr>
<tr>
<td>POS no Context 0.05</td>
<td>19.48</td>
<td>26.69</td>
</tr>
<tr>
<td>POS no Context 0.1</td>
<td>19.55</td>
<td>26.46</td>
</tr>
<tr>
<td>POS no Context 0.2</td>
<td>19.30</td>
<td>26.01</td>
</tr>
<tr>
<td>POS no Context 0.3</td>
<td>19.22</td>
<td>25.73</td>
</tr>
<tr>
<td>POS + Context 0.01</td>
<td>19.34</td>
<td>25.85</td>
</tr>
<tr>
<td>POS + Context 0.05</td>
<td>19.34</td>
<td>25.86</td>
</tr>
<tr>
<td>POS + Context 0.1</td>
<td>19.44</td>
<td>25.79</td>
</tr>
<tr>
<td>unseen Baseline (RO3)</td>
<td>17.69</td>
<td>23.70</td>
</tr>
<tr>
<td>unseen no Context</td>
<td>17.78</td>
<td>24.79</td>
</tr>
<tr>
<td>unseen with Context</td>
<td>17.79</td>
<td>23.87</td>
</tr>
</tbody>
</table>
## Results

Rottmann, Vogel (2007)

<table>
<thead>
<tr>
<th>System</th>
<th>en $\rightarrow$ es</th>
<th>System</th>
<th>en $\rightarrow$ de</th>
<th>de $\rightarrow$ en</th>
</tr>
</thead>
<tbody>
<tr>
<td>unseen Baseline (RO3)</td>
<td>48.51</td>
<td>unseen Baseline (RO3)</td>
<td>17.69</td>
<td>23.70</td>
</tr>
<tr>
<td>unseen no Context</td>
<td>49.52</td>
<td>unseen no Context</td>
<td>17.78</td>
<td>24.79</td>
</tr>
<tr>
<td>unseen with Context</td>
<td>49.49</td>
<td>unseen with Context</td>
<td>17.79</td>
<td>23.87</td>
</tr>
<tr>
<td>unseen combination</td>
<td>49.58</td>
<td>unseen combination</td>
<td>18.27</td>
<td>24.85</td>
</tr>
<tr>
<td>unseen combination-Lex</td>
<td>49.83</td>
<td>unseen combination-Lex</td>
<td>18.21</td>
<td>24.88</td>
</tr>
</tbody>
</table>
# Results

Rottmann, Vogel (2007)

<table>
<thead>
<tr>
<th>Corpus</th>
<th>en → de</th>
<th>de → en</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combination</td>
<td>19.61</td>
<td>26.88</td>
</tr>
<tr>
<td>Reordered (Giza)</td>
<td>19.44</td>
<td>26.76</td>
</tr>
<tr>
<td>Reordered (Lattice)</td>
<td>20.00</td>
<td>27.06</td>
</tr>
<tr>
<td>unseen Baseline (RO3)</td>
<td>17.69</td>
<td>23.70</td>
</tr>
<tr>
<td>unseen combination</td>
<td>18.27</td>
<td>24.85</td>
</tr>
<tr>
<td>unseen reordered corpus</td>
<td>18.42</td>
<td>25.06</td>
</tr>
</tbody>
</table>
Introduction

Reordering in Phrase-based SMT
Word Orders between Language Pairs

POS-based Reordering

Popović, Ney (2006)
Crego, Mariño (2006)
Rottmann, Vogel (2007)

Syntactic Reordering

Nguyen, Shimazu (2006)
Li et al. (2007)

Conclusion
Languages: English, French.
Learn rewrite patterns for transformation of parse trees of source sentences.
Newly ordered source sentence sent to the decoder.
A rewrite Pattern is a tuple: (Src Rule, Tgt Rule, Src Head Position, Tgt Head Position, Child-Alignment).

If the Src or Tgt Rule contains the head word, the pattern is said to be lexicalized. (Useful e.g. french adjectives)
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
  - Let S be source phrase, T be target phrase.
  - $Score(S, T) = \frac{\text{links}(S, T)}{\text{Span}(S) + \text{Span}(T)}$
  - Align S to the T that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
  - Find if children on src and target align among themselves
  - Find if source head node aligns to target head node
  - Source node is lexicalized iff Target node is too.
  - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
  - Let $S$ be source phrase, $T$ be target phrase.
  - $Score(S, T) = \frac{\text{links}(S, T)}{\text{Span}(S) + \text{Span}(T)}$
  - Align $S$ to the $T$ that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
  - Find if children on src and target align among themselves
  - Find if source head node aligns to target head node
  - Source node is lexicalized iff Target node is too.
  - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
    - Let S be source phrase, T be target phrase.
    - \[ \text{Score}(S, T) = \frac{\text{links}(S, T)}{\text{Span}(S)+\text{Span}(T)} \]
    - Align S to the T that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
  - Find if children on src and target align among themselves
  - Find if source head node aligns to target head node
  - Source node is lexicalized iff Target node is too.
  - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
  - Let S be source phrase, T be target phrase.
    - Score(S, T) = \( \frac{\text{links}(S,T)}{\text{Span}(S)+\text{Span}(T)} \)
    - Align S to the T that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
  - Find if children on src and target align among themselves
  - Find if source head node aligns to target head node
  - Source node is lexicalized iff Target node is too.
  - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
  - Let $S$ be source phrase, $T$ be target phrase.
  - $\text{Score}(S, T) = \frac{\text{links}(S, T)}{\text{Span}(S) + \text{Span}(T)}$
  - Align $S$ to the $T$ that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
  - Find if children on src and target align among themselves
  - Find if source head node aligns to target head node
  - Source node is lexicalized iff Target node is too.
  - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
  - Let S be source phrase, T be target phrase.
  - \[ \text{Score}(S, T) = \frac{\text{links}(S, T)}{\text{Span}(S) + \text{Span}(T)} \]
  - Align S to the T that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
  - Find if children on src and target align among themselves
  - Find if source head node aligns to target head node
  - Source node is lexicalized iff Target node is too.
  - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
  - Let S be source phrase, T be target phrase.
  - \[
  \text{Score}(S, T) = \frac{\text{links}(S, T)}{\text{Span}(S) + \text{Span}(T)}
  \]
  - Align S to the T that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
  - Find if children on src and target align among themselves
  - Find if source head node aligns to target head node
  - Source node is lexicalized iff Target node is too.
  - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
  - Let S be source phrase, T be target phrase.
  - \( \text{Score}(S, T) = \frac{\text{links}(S, T)}{\text{Span}(S) + \text{Span}(T)} \)
  - Align S to the T that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
    - Find if children on src and target align among themselves
    - Find if source head node aligns to target head node
    - Source node is lexicalized iff Target node is too.
    - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
  - Let S be source phrase, T be target phrase.
  - \( \text{Score}(S, T) = \frac{\text{links}(S, T)}{\text{Span}(S) + \text{Span}(T)} \)
  - Align S to the T that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
  - Find if children on src and target align among themselves
  - Find if source head node aligns to target head node
  - Source node is lexicalized iff Target node is too.
  - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
  - Let S be source phrase, T be target phrase.
  - \[ \text{Score}(S, T) = \frac{\text{links}(S, T)}{\text{Span}(S) + \text{Span}(T)} \]
  - Align S to the T that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
  - Find if children on src and target align among themselves
  - Find if source head node aligns to target head node
  - Source node is lexicalized iff Target node is too.
  - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
  - Let S be source phrase, T be target phrase.
  - \[ \text{Score}(S, T) = \frac{\text{links}(S,T)}{\text{Span}(S)+\text{Span}(T)} \]
  - Align S to the T that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
  - Find if children on src and target align among themselves
  - Find if source head node aligns to target head node
  - Source node is lexicalized iff Target node is too.
  - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
  - Let S be source phrase, T be target phrase.
  - \[ \text{Score}(S, T) = \frac{\text{links}(S, T)}{\text{Span}(S) + \text{Span}(T)} \]
  - Align S to the T that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
  - Find if children on src and target align among themselves
  - Find if source head node aligns to target head node
  - Source node is lexicalized iff Target node is too.
  - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
  - Let S be source phrase, T be target phrase.
  - \( \text{Score}(S, T) = \frac{\text{links}(S, T)}{\text{Span}(S) + \text{Span}(T)} \)
  - Align S to the T that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
  - Find if children on src and target align among themselves
  - Find if source head node aligns to target head node
  - Source node is lexicalized iff Target node is too.
  - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
  - Let S be source phrase, T be target phrase.
  - \[ \text{Score}(S, T) = \frac{\text{links}(S,T)}{\text{Span}(S)+\text{Span}(T)} \]
  - Align S to the T that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
  - Find if children on src and target align among themselves
  - Find if source head node aligns to target head node
  - Source node is lexicalized iff Target node is too.
  - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
Learning Rewrite patterns

- Parse the sentences. They used Slot Grammar parsers.
- Align Phrases.
  - How?
  - Let S be source phrase, T be target phrase.
  - \[ \text{Score}(S, T) = \frac{\text{links}(S, T)}{\text{Span}(S) + \text{Span}(T)} \]
  - Align S to the T that gives the max score.
- Extract Rewrite Patterns.
  - Find aligned nodes
  - Find if children on src and target align among themselves
  - Find if source head node aligns to target head node
  - Source node is lexicalized iff Target node is too.
  - Rule-length must be atmost 5
- Apply Rewrite Patterns
  - Use Greedy Strategy
  - Visit each node and apply the most specific pattern applicable at that node.
En-Fr Canadian Hansard corpus used (90M word)
2.9M rewrite patterns extracted
Patterns filtered down to 56K.
1042 patterns were lexicalized

<table>
<thead>
<tr>
<th>rewrite patterns</th>
<th>non-monotonic decoding</th>
<th>monotonic decoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>not used</td>
<td>0.187</td>
<td>0.196</td>
</tr>
<tr>
<td>used</td>
<td>0.185</td>
<td>0.215</td>
</tr>
</tbody>
</table>
Outline

1. Introduction
   - Reordering in Phrase-based SMT
   - Word Orders between Language Pairs

2. POS-based Reordering
   - Popović, Ney (2006)
   - Crego, Mariño (2006)
   - Rottmann, Vogel (2007)

3. Syntactic Reordering
   - Nguyen, Shimazu (2006)
   - Li et al. (2007)

4. Conclusion
Works with German. German has more reordering phenomena than French wrt English.

Rules are manually crafted. Not automatically learned.

Rule-set consists of 6 transformations, very specific to German.
Europarl Corpus (750K sentences) used.
Baseline SMT System score: 25.2
New System BLEU Score: 26.8
Human Evaluation performed on 100 random sentences by 2 judges
- 33 sentences showed improvement over SMT
- 13 sentences were worse after reordering
Outline

1. Introduction
   - Reordering in Phrase-based SMT
   - Word Orders between Language Pairs

2. POS-based Reordering
   - Popović, Ney (2006)
   - Crego, Mariño (2006)
   - Rottmann, Vogel (2007)

3. Syntactic Reordering
   - Nguyen, Shimazu (2006)
   - Li et al. (2007)

4. Conclusion
Syntactic Transformational Model for SMT
Nguyen, Shimazu (2006)

- Unlike previous syntactic methods, this transformational model is based on statistical decisions.
- Rules are learned from corpora, and scored.
- Application of rules to new sentences is also done statistically.
What is a Transformation?
Nguyen, Shimazu (2006)

- There could be multiple ways to reorder a CFG rule.
- Lexicalization of rules can help decide which reordering should be applied.
- Lexicalization can lead to too many rules: score estimation is a problem.
- Use LPCFG to get the scores
The Training process
Nguyen, Shimazu (2006)

- Parse text. Get GIZA alignments.
- Align source-side phrases. (Similar to Xia, McCord (2004))
- If there are one-to-many alignments:
  - If source span is one word, choose the best link based on intersection of bidirectional alignments and lexical scores.
  - For each word outside source phrase, there should be no link to any word outside the target phrase, and vice versa.
- For each node, based on the target phrase position of children, learn a reordering rule.
- Score all rules:
  \[
p(LHS \rightarrow RHS | LHS \rightarrow RHS') = \frac{n(LHS \rightarrow RHS | LHS \rightarrow RHS')}{n(LHS \rightarrow RHS')}
\]
The Training process
Nguyen, Shimazu (2006)

- Parse text. Get GIZA alignments.
- Align source-side phrases. (Similar to Xia, McCord (2004))
- If there are one-to-many alignments:
  - If source span is one word, choose the best link based on intersection of bidirectional alignments and lexical scores.
  - For each word outside source phrase, there should be no link to any word outside the target phrase, and vice versa.
- For each node, based on the target phrase position of children, learn a reordering rule.
- Score all rules:
  \[
p(LHS \rightarrow RHS | LHS \rightarrow RHS') = \frac{n(LHS \rightarrow RHS | LHS \rightarrow RHS')}{n(LHS \rightarrow RHS')}.\]
The Training process
Nguyen, Shimazu (2006)

- Parse text. Get GIZA alignments.
- Align source-side phrases. (Similar to Xia, McCord (2004))
- If there are one-to-many alignments:
  - If source span is one word, choose the best link based on intersection of bidirectional alignments and lexical scores.
  - For each word outside source phrase, there should be no link to any word outside the target phrase, and vice versa.
- For each node, based on the target phrase position of children, learn a reordering rule.
- Score all rules:
  \[ p(LHS \rightarrow RHS | LHS \rightarrow RHS') = \frac{n(LHS \rightarrow RHS | LHS \rightarrow RHS')}{n(LHS \rightarrow RHS')} . \]
The Training process
Nguyen, Shimazu (2006)

- Parse text. Get GIZA alignments.
- Align source-side phrases. (Similar to Xia, McCord (2004))
- If there are one-to-many alignments:
  - If source span is one word, choose the best link based on intersection of bidirectional alignments and lexical scores.
  - For each word outside source phrase, there should be no link to any word outside the target phrase, and vice versa.
- For each node, based on the target phrase position of children, learn a reordering rule.
- Score all rules:
  \[ p(LHS \rightarrow RHS | LHS \rightarrow RHS') = \frac{n(LHS \rightarrow RHS | LHS \rightarrow RHS')}{n(LHS \rightarrow RHS')} . \]
The Training process
Nguyen, Shimazu (2006)

- Parse text. Get GIZA alignments.
- Align source-side phrases. (Similar to Xia, McCord (2004))
- If there are one-to-many alignments:
  - If source span is one word, choose the best link based on intersection of bidirectional alignments and lexical scores.
  - For each word outside source phrase, there should be no link to any word outside the target phrase, and vice versa.
- For each node, based on the target phrase position of children, learn a reordering rule.
- Score all rules:
  \[ p(LHS \rightarrow RHS|LHS \rightarrow RHS') = \frac{n(LHS \rightarrow RHS|LHS \rightarrow RHS')}{n(LHS \rightarrow RHS')} \]
The Training process
Nguyen, Shimazu (2006)

- Parse text. Get GIZA alignments.
- Align source-side phrases. (Similar to Xia, McCord (2004))
- If there are one-to-many alignments:
  - If source span is one word, choose the best link based on intersection of bidirectional alignments and lexical scores.
  - For each word outside source phrase, there should be no link to any word outside the target phrase, and vice versa.
- For each node, based on the target phrase position of children, learn a reordering rule.

Score all rules:
\[ p(LHS \rightarrow RHS|LHS \rightarrow RHS') = \frac{n(LHS \rightarrow RHS|LHS \rightarrow RHS')}{n(LHS \rightarrow RHS')} \]
The Training process
Nguyen, Shimazu (2006)

- Parse text. Get GIZA alignments.
- Align source-side phrases. (Similar to Xia, McCord (2004))
- If there are one-to-many alignments:
  - If source span is one word, choose the best link based on intersection of bidirectional alignments and lexical scores.
  - For each word outside source phrase, there should be no link to any word outside the target phrase, and vice versa.
- For each node, based on the target phrase position of children, learn a reordering rule.
- Score all rules:
  \[ p(LHS \rightarrow RHS | LHS \rightarrow RHS') = \frac{n(LHS \rightarrow RHS | LHS \rightarrow RHS')} {n(LHS \rightarrow RHS')} \]
Applying the rules
Nguyen, Shimazu (2006)

- Parse the input sentence.
- Lexicalize the tree. (Propogate heads bottom up).
- For the tree, apply the best possible transformation sequence.

\[
Q^* = \{ RS_i^* : RS_i^* = \arg\max [P(L_i \rightarrow R_i | L_i \rightarrow R'_i) \times P(L_i \rightarrow R'_i)] \}
\]
- Extract the surface string
Applying the rules
Nguyen, Shimazu (2006)

- Parse the input sentence.
- Lexicalize the tree. (Propogate heads bottom up).
- For the tree, apply the best possible transformation sequence.
- \( Q^* = \{ RS^*_i : RS^*_i = \arg\max [P(L_i \rightarrow R_i|L_i \rightarrow R'_i) \ast P(L_i \rightarrow R'_i)] \} \)
- Extract the surface string
Applying the rules
Nguyen, Shimazu (2006)

- Parse the input sentence.
- Lexicalize the tree. (Propagate heads bottom up).
- For the tree, apply the best possible transformation sequence.
  
  \[ Q^* = \{ RS_i^* : RS_i^* = \text{argmax} \left[ P(L_i \rightarrow R_i | L_i \rightarrow R'_i) \times P(L_i \rightarrow R'_i) \right] \} \]

- Extract the surface string
Applying the rules
Nguyen, Shimazu (2006)

- Parse the input sentence.
- Lexicalize the tree. (Propogate heads bottom up).
- For the tree, apply the best possible transformation sequence.

\[ Q^* = \{ RS_i^* : RS_i^* = \arg\max \left[P(L_i \rightarrow R_i | L_i \rightarrow R'_i) \ast P(L_i \rightarrow R'_i)\right] \} \]

- Extract the surface string
Applying the rules
Nguyen, Shimazu (2006)

- Parse the input sentence.
- Lexicalize the tree. (Propogate heads bottom up).
- For the tree, apply the best possible transformation sequence.
- \( Q^* = \{ RS_i^* : RS_i^* = \arg\max \left[ P(L_i \rightarrow R_i | L_i \rightarrow R_i') \star P(L_i \rightarrow R_i') \right] \} \)
- Extract the surface string
Experiments
Nguyen, Shimazu (2006)

- Experimented with English, Vietnamese, French
- Restricted training to 40K trees.
Experiments
Nguyen, Shimazu (2006)

- Experimented with English, Vietnamese, French
- Restricted training to 40K trees.
Experiments
Nguyen, Shimazu (2006)

- Experimented with English, Vietnamese, French
- Restricted training to 40K trees.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>UCFGRs</th>
<th>TRGs</th>
<th>AGs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer</td>
<td>4779</td>
<td>3702</td>
<td>951</td>
</tr>
<tr>
<td>Conversation</td>
<td>3634</td>
<td>2642</td>
<td>669</td>
</tr>
<tr>
<td>Europarl</td>
<td>14462</td>
<td>10738</td>
<td>3706</td>
</tr>
</tbody>
</table>
Results
Nguyen, Shimazu (2006)

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Baseline</th>
<th>Syntactic transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer</td>
<td>45.12</td>
<td>47.62</td>
</tr>
<tr>
<td>Conversation</td>
<td>33.85</td>
<td>36.26</td>
</tr>
<tr>
<td>Europarl</td>
<td>26.41</td>
<td>28.02</td>
</tr>
</tbody>
</table>
## Results

Nguyen, Shimazu (2006)

<table>
<thead>
<tr>
<th>Maximum phrase size</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pharaoh</td>
<td>21.71</td>
<td>24.84</td>
<td>25.74</td>
<td>26.19</td>
<td>26.41</td>
</tr>
<tr>
<td>Syntactic transformation</td>
<td>24.1</td>
<td>27.01</td>
<td>27.74</td>
<td>27.88</td>
<td>28.02</td>
</tr>
</tbody>
</table>
## Results

Nguyen, Shimazu (2006)

<table>
<thead>
<tr>
<th>Training-set size</th>
<th>10K</th>
<th>20K</th>
<th>40K</th>
<th>80K</th>
<th>94K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pharaoh</td>
<td>21.84</td>
<td>23.35</td>
<td>24.43</td>
<td>25.43</td>
<td>25.74</td>
</tr>
<tr>
<td>Syntactic transformation</td>
<td>23.65</td>
<td>25.67</td>
<td>26.86</td>
<td>27.52</td>
<td>27.74</td>
</tr>
</tbody>
</table>
## Results

Nguyen, Shimazu (2006)

<table>
<thead>
<tr>
<th>Training-set size</th>
<th>10K</th>
<th>20K</th>
<th>40K</th>
<th>80K</th>
<th>94K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pharaoh</td>
<td>1.98</td>
<td>2.52</td>
<td>2.93</td>
<td>3.45</td>
<td>3.67</td>
</tr>
<tr>
<td>Syntactic transformation</td>
<td>0.1</td>
<td>0.13</td>
<td>0.16</td>
<td>0.19</td>
<td>0.22</td>
</tr>
</tbody>
</table>
Outline

1. Introduction
   - Reordering in Phrase-based SMT
   - Word Orders between Language Pairs

2. POS-based Reordering
   - Popović, Ney (2006)
   - Crego, Mariño (2006)
   - Rottmann, Vogel (2007)

3. Syntactic Reordering
   - Nguyen, Shimazu (2006)
   - Li et al. (2007)

4. Conclusion
Probabilistic approach to Syntax-based RO for SMT
Li et al. (2007)

- Previous syntactic systems propose: \( S \rightarrow S' \rightarrow T \).
- They propose: \( S \rightarrow n \ast S' \rightarrow n \ast T \rightarrow \hat{T} \).
- Give up using rewrite patterns. Instead acquire RO knowledge.
Training
Li et al. (2007)

- Simplified case: binary tree.
- Let $A \rightarrow B C$ be a node in the tree.
- Use word alignments to determine:
  1. What is the minimum and maximum position on target side that yield of $B$ aligns to? ($T(B)$)
  2. What is the minimum and maximum position on target side that yield of $C$ aligns to? ($T(C)$)
- If $T(B)$ and $T(C)$ overlap:
  1. Keep remove the worst-scoring link from word-alignments in the phrases until overlap goes away.
  2. If too many links are removed, the node $A$ not used as training item.
- Easily extended to n-ary trees.
Strategy 1: Learn Rules

1. Consider every rule $Z : XY$ in the trees
2. Use relative frequency to estimate how many times it is reordered

Strategy 2: Maximum Entropy Model:

1. Binary classification of whether the children of a node are reordered or not.
2. Features used:
   - Leftmost word of a phrase and its POS
   - Rightmost word of a phrase and its POS
   - Head word and its POS
   - Context words (phrase $\pm 1$ word) and their POS.
Applying learned knowledge
Li et al. (2007)

- Use bottom-up approach
- If current node has unary production, assign it a score of 1.
- Determine which rules are applicable at the node, or determine via EM if node should be reordered. Obtain the rule score of the new order.
- Set value of current node: \( val = \text{RuleScore} \times \text{Product of Values of Children} \).
- Keep track of N-highest probabilities of nodes. They correspond to the N-Best list.
During Decoding
Li et al. (2007)

- Split input sentence into clauses, using IP nodes in parse trees.
- Reorder each clause, get an n-best list for each clause.
- Translate each of the n-best items of each clause
- Choose best-scoring translation of each clause
- Combine these translations back to one sentence.
- Decoder has additional feature: \( P(S \rightarrow S') \). This is the score of the tree for each reordering.
Experiments
Li et al. (2007)

- Pharaoh-like decoder
- GIGAword corpus as training data
- MT05 Chinese-English data for testing.
Experiments
Li et al. (2007)

- Pharaoh-like decoder
- GIGAword corpus as training data
- MT05 Chinese-English data for testing.
Experiments
Li et al. (2007)

- Pharaoh-like decoder
- GIGAword corpus as training data
- MT05 Chinese-English data for testing.
Experiments
Li et al. (2007)

- Pharaoh-like decoder
- GIGAword corpus as training data
- MT05 Chinese-English data for testing.

<table>
<thead>
<tr>
<th>Branching Factor</th>
<th>2</th>
<th>3</th>
<th>&gt;3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>12294</td>
<td>3173</td>
<td>1280</td>
</tr>
<tr>
<td>Percentage</td>
<td>73.41</td>
<td>18.95</td>
<td>7.64</td>
</tr>
</tbody>
</table>
## Results
Li et al. (2007)

<table>
<thead>
<tr>
<th>Test</th>
<th>Setting</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2,ary</td>
</tr>
<tr>
<td>B1</td>
<td>standard phrase-based SMT</td>
<td>29.22</td>
</tr>
<tr>
<td>B2</td>
<td>(B1) + clause splitting</td>
<td>29.13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test</th>
<th>Setting</th>
<th>BLEU 2-ary</th>
<th>BLEU 2,3-ary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>rule</td>
<td>29.77</td>
<td>30.31</td>
</tr>
<tr>
<td>2</td>
<td>ME (phrase label)</td>
<td>29.93</td>
<td>30.49</td>
</tr>
<tr>
<td>3</td>
<td>ME (left,right)</td>
<td>30.10</td>
<td>30.53</td>
</tr>
<tr>
<td>4</td>
<td>ME ((3)+head)</td>
<td>30.24</td>
<td>30.71</td>
</tr>
<tr>
<td>5</td>
<td>ME ((3)+phrase label)</td>
<td>30.12</td>
<td>30.30</td>
</tr>
<tr>
<td>6</td>
<td>ME ((4)+context)</td>
<td>30.24</td>
<td>30.76</td>
</tr>
<tr>
<td>Test</td>
<td>Setting</td>
<td>BLEU</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>--------------------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>length constraint</td>
<td>30.52</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>DL=0</td>
<td>30.48</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>n=100</td>
<td>30.78</td>
<td></td>
</tr>
</tbody>
</table>
Bright future with forests
Almost all systems currently use rules, or patterns to transform text

- It would be interesting to use wildcards in rules
- For POS, this may be easier to define: AUX-V-* becomes AUX-*-V
- Wildcards with trees could be really wild!
- Recent experiments with reordering arabic trees: ‘good’ rules don’t just contain wildcards, they contain tgrep style regular expressions.
- Learning such rules is a very challenging problem.
Almost all systems currently use rules, or patterns to transform text.

It would be interesting to use wildcards in rules.

For POS, this may be easier to define: AUX-V-* becomes AUX-*-V.

Wildcards with trees could be really wild!

Recent experiments with reordering arabic trees: ‘good’ rules don’t just contain wildcards, they contain tgrep style regular expressions.

Learning such rules is a very challenging problem.
Wildcard rules

- Almost all systems currently use rules, or patterns to transform text
- It would be interesting to use wildcards in rules
- For POS, this may be easier to define: AUX-V-* becomes AUX-*-V
- Wildcards with trees could be really wild!
- Recent experiments with reordering arabic trees: ‘good’ rules don’t just contain wildcards, they contain tgrep style regular expressions.
- Learning such rules is a very challenging problem.
Almost all systems currently use rules, or patterns to transform text.

It would be interesting to use wildcards in rules.

For POS, this may be easier to define: `AUX-V-*` becomes `AUX-*-V`.

Wildcards with trees could be really wild!

Recent experiments with reordering arabic trees: ‘good’ rules don’t just contain wildcards, they contain `tgrep` style regular expressions.

Learning such rules is a very challenging problem.
Wildcard rules

Almost all systems currently use rules, or patterns to transform text.

It would be interesting to use wildcards in rules.

For POS, this may be easier to define: AUX-V-* becomes AUX-*-V.

Wildcards with trees could be really wild!

Recent experiments with reordering arabic trees: ‘good’ rules don’t just contain wildcards, they contain tgrep style regular expressions.

Learning such rules is a very challenging problem.
Almost all systems currently use rules, or patterns to transform text.

It would be interesting to use wildcards in rules.

For POS, this may be easier to define: AUX-V-* becomes AUX-*-V.

Wildcards with trees could be really wild!

Recent experiments with reordering arabic trees: ‘good’ rules don’t just contain wildcards, they contain tgrep style regular expressions.

Learning such rules is a very challenging problem.
Parser issues

- Syntax-based rules depend on what parser is used.
- Stanford parser creates deep trees. Too many nodes may hamper the process of learning good rules. Shallow parsers may be an option.
- Not all languages have a parser trained on large data. Does a light-weight parser introduce too much noise in the forest? Are current methods robust to parser errors?
Syntax-based rules depend on what parser is used.

Stanford parser creates deep trees. Too many nodes may hamper the process of learning good rules. Shallow parsers may be an option.

Not all languages have a parser trained on large data. Does a light-weight parser introduce too much noise in the forest? Are current methods robust to parser errors?
Parser issues

- Syntax-based rules depend on what parser is used.
- Stanford parser creates deep trees. Too many nodes may hamper the process of learning good rules. Shallow parsers may be an option.
- Not all languages have a parser trained on large data. Does a light-weight parser introduce too much noise in the forest? Are current methods robust to parser errors?
Evaluation issues

- Is BLEU a good metric to study effect of reordering sentences?
- If source-reordering claims to produce more grammatical sentences, could the grammaticality be evaluated?
Evaluation issues

- Is BLEU a good metric to study effect of reordering sentences?
- If source-reordering claims to produce more grammatical sentences, could the grammaticality be evaluated?
Summary

Works great the reordering of sentences source. The problem but solved is not.

Questions

Search Questions on Google Images; Feel Lucky