Source-side Dependency Tree Reordering Models with Subtree Movements and Constraints

Nguyen Bach, Qin Gao and Stephan Vogel
Carnegie Mellon University
Overview

• We introduce **source-side dependency tree reordering models**
  
  • Inspired by **lexicalized reordering model** (Koehn et. al 2005), **hierarchical dependency translation** (Shen et. al, 2008) and **cohesive decoding** (Cherry, 2008)

• We model reordering events of phrases associated with source-side dependency trees
  
  • **Inside/Outside subtree movements** efficiently capture the statistical distribution of the subtree-to-subtree transitions in training data

  • Utilize subtree movements directly at the decoding time alongside with **cohesive constraints** to guide the search process

• Improvements are shown in English-Spanish and English-Iraqi tasks
Outline

• **Background & Motivations**
  • Source-side dependency tree reordering models
    – Modeling
    – Training
    – Decoding

• **Experiments & Analysis**

• **Conclusions**
Background of Reordering Models

- Explicitly model phrase reordering distances
- Put syntactic analysis of the target language into both modeling and decoding
- Use source language syntax
Reordering Models

- Explicitly model phrase reordering distances
  - Distance-based (Och, 2002; Koehn et al., 2003)
  - Lexicalized phrase (Tillmann, 2004; Koehn et al., 2005; Al-Onaizan and Papineni, 2006)
  - Hierarchical phrase (Galley and Manning, 2008)
  - MaxEnt classifier (Zens and Ney, 2006; Xiong, et al., 2006; Chang, et al., 2009)

- Put syntactic analysis of the target language into both modeling and decoding
  - Direct model target language constituents movement in either constituency trees (Yamada and Knight, 2001; Galley et al., 2006; Zollmann et al., 2008) or dependency trees (Quirk, et al., 2005)
  - Hierarchical phrase-based (Chiang, 2005; Shen et al., 2008)

- Use source language syntax
  - Preprocessing with syntactic reordering rules (Xia and McCord, 2004; Collins et al., 2005; Rottmann and Vogel, 2007; Wang et al., 2007; Xu et al., 2009)
  - Use syntactical analysis to provide multiple source sentence reordering options through word lattices (Zhang et al., 2007; Li et al., 2007; Elming, 2008).
Reordering Models

- Explicitly model phrase reordering distances
- Put syntactic analysis of the target language into both modeling and decoding
- Use source language syntax

Source-side

- Distance-based (Och, 2002; Koehn et al., 2003)
- Lexicalized phrase (Tillmann, 2004; Koehn et al., 2005; Al-Onaizan and Papineni, 2006)
- Hierarchical phrase (Galley and Manning, 2008)

Dependency Tree

- Direct modeling of target language constituents movement in either constituency trees (Yamada and Knight, 2001; Galley et al., 2006; Zollmann et al., 2008) or dependency trees (Quirk, et al., 2005)

Reordering Models

- Hierarchical phrase-based (Chiang, 2005; Shen et al., 2008)

with Subtree Movements and Constraints

Preprocessing with syntactic reordering rules (Xia and McCord, 2004; Collins et al., 2005; Rottmann and Vogel, 2007; Wang et al., 2007; Xu et al., 2009)

Use syntactical analysis to provide multiple source sentence reordering options through word lattices (Zhang et al., 2007; Li et al., 2007; Elming, 2008).
What are the differences?

• Instead of using flat word structures to extract reordering events, utilize **source-side dependency structures**
  – Provide more linguistic cues for reordering events

• Instead of using pre-defined reordering patterns, learn **reordering feature distributions** from training data
  – Capture reordering events from real data

• Instead of preprocessing the data, **discriminatively train** the reordering model via MERT
  – Tighter integration with the decoder
Cohesive Decoding

• A cohesive decoding (Cherry, 08; Bach et. al., 09) is forcing the cohesive constraint:
  – *When the decoder begins translation any part of a source subtree, it must cover all words under that subtree before it can translate anything outside.*

• Source-side dependency tree reordering models
  – Efficiently capture the statistical distribution of the subtree-to-subtree transitions in training data.
  – Directly utilize it at the decoding time to guide the search process.
Outline

• Background of Reordering Models
• **Source-side dependency tree reordering models**
  – Modeling
  – Training
  – Decoding
• Experiments & Analysis
• Conclusions
Lexicalized Reordering Models (Tillmann, 2004; Koehn, et.al., 2005; Al-Onaizan & Papineni, 2006)

\[ p(O \mid e, f) = \prod_{i=1}^{n} p(o_i \mid \bar{e}_i, \bar{f}_{a_i}, a_{i-1}, a_i) \]

where

- \( f \) is the input sentence;
- \( e = (\bar{e}_1, ..., \bar{e}_n) \) is the target language phrases;
- \( a = (a_1, ..., a_n) \) is phrase alignments;
- \( \bar{f}_{a_i} \) is a source phrase which has a translated phrase \( \bar{e}_i \) defined by an alignment \( a_i \);
- \( O \) is orientation phrase sequence; each \( o_i \) has a value over 3 possibles (M, S, D);
<table>
<thead>
<tr>
<th>16</th>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>would</td>
<td>therefore</td>
<td>once</td>
<td>more</td>
<td>ask</td>
<td>you</td>
<td>to</td>
<td>ensure</td>
<td>that</td>
<td>we</td>
<td>get</td>
<td>a</td>
<td>Dutch</td>
<td>channel</td>
<td>as</td>
<td>well</td>
</tr>
</tbody>
</table>

**Quisiera**

**Tanto**

**Lo**

**Por**
I would therefore once more ask you to ensure that we get a Dutch channel as well.
I would therefore ask you to ensure that we get a Dutch channel as well.
Pros & Cons of Lexicalized Reordering Models

• Pros
  – intuitively model flat word movements
  – well-defined for phrase-based framework

• Cons
  – No linguistics structures
  – Need alignment matrix to determine movements
Completed/Open subtrees

A completed subtree

All words under a node have been translated then we call a completed subtree
Completed/Open subtrees

A subtree that has begun translation but not yet complete, an open subtree
Inside/Outside subtree movements

“c” is moving **inside** a subtree rooted at “b”

A structure is moving **inside** a subtree if it helps the subtree to be completed or less open.
A structure is moving outside a subtree if it leaves the subtree to be open.

“d e” is moving outside a subtree rooted at “b”
Source-side Dependency Tree (SDT) Reordering Models

\[ p(D | e, f) = \prod_{i=1}^{n} p(d_i | \bar{e}_i, \bar{f}_{a_i}, a_i, s_{i-1}, s_i) \]

**where**

- \( f \) is the input sentence;
- \( e = (\bar{e}_1, ..., \bar{e}_n) \) is the target language phrases;
- \( a = (a_1, ..., a_n) \) is phrase alignments;
- \( \bar{f}_{a_i} \) is a source phrase which has a translated phrase \( \bar{e}_i \) defined by an alignment \( a_i \);

\( s_i \) and \( s_{i-1} \) are dependency structures of source phrases \( \bar{f}_{a_i} \) and \( \bar{f}_{a_{i-1}} \);

\( D \) represents the sequence of syntactic phrase movements over source dependency tree;

each \( d_i = \{I, O\} \);
I would therefore ask you to ensure that we get a Dutch channel as well.
I therefore once ask more ensure you to that we get channel a Dutch as well
Once we ask you to ensure that we get a Dutch channel as well, we would insist on it, and therefore I would ensure that we get it inside.
Pedite would ask you to ensure that we get a Dutch channel as well.

Pedite would ask you to ensure that we get a Dutch channel as well.
I would ask you to ensure that we get a Dutch channel as well.
I therefore once ask more ensure to that we get channel as Dutch well.

Source-side Dependency Tree R.M.

Inside Outside Inside Inside

Lexicalized R.M.

Discontinuous Swap Discontinuous Monotone
Extended Source-side Dependency Tree (SDT) Reordering Models

\[ p(D | e, f) = \prod_{i=1}^{n} p((o_d)_i | \bar{e}_i, \bar{f}_{a_i}, a_i, a_{i-1}, s_{i-1}, s_i) \]

where

- \( f \) is the input sentence;
- \( e = (\bar{e}_1, \ldots, \bar{e}_n) \) is the target language phrases;
- \( a = (a_1, \ldots, a_n) \) is phrase alignments;
- \( \bar{f}_{a_i} \) is a source phrase which has a translation \( \bar{e}_i \) defined by an alignment \( a_i \);
- \( s_i \) and \( s_{i-1} \) are dependency structures of source phrases \( \bar{f}_{a_i} \) and \( \bar{f}_{a_{i-1}} \);

\( D \) represents the sequence of syntactic phrase movements over source dependency tree;

each \((o_d)_i = \{M_I, S_I, D_I, M_O, S_O, D_O\}\).
Extended Source-side Dependency Tree (SDT) Reordering Models

\[ p(D | e, a, \bar{f}_s) = \prod_{i=1}^{n} D_i \]

where \( f \) is the phrase source dependency tree,
\( e = (e_1, \ldots, e_s) \),
\( a = (a_1, \ldots, a_s) \),
\( \bar{f}_s \) is a sequence of \( s \) trees,
\( s_i \) and \( s_i \) are input sentence alignments.

\( D \) represents reordering movements over source dependency tree;

each \((o_d)_i = \{M_I, S_I, D_I, M_O, S_O, D_O\};\)
Training

• Obtain dependency parse of the source side
• Given a sentence pair and the source side dependency tree
  – Phrase extraction: also extract source dependency structures of phrase pairs
  – Identify Inside/Outside movement by using Interruption Check Algorithms (Bach et.al., 2009)
Training

**DO:** a **joint probability** of subtree movements and lexicalized orientations

\[
p((o_j - d_k) | \bar{e}_i, \bar{f}_{a_i}, o_j, d_k) = \frac{\text{count} \ (o_j - d_k) + \gamma}{\sum_k \sum_j (\text{count} \ (o_j - d_k) + \gamma)}
\]

**DOD:** **conditioned** on subtree movements

\[
p((o_j - d_k) | \bar{e}_i, \bar{f}_{a_i}, d_k) = \frac{\text{count} \ (o_j - d_k) + \gamma}{\sum_k (\text{count} \ (o_j - d_k) + \gamma)}
\]

**DOO:** **conditioned** on lexicalized orientations

\[
p((o_j - d_k) | \bar{e}_i, \bar{f}_{a_i}, o_j) = \frac{\text{count} \ (o_j - d_k) + \gamma}{\sum_j (\text{count} \ (o_j - d_k) + \gamma)}
\]
Decoding

• Without cohesive constraints
  – Having no information about the source dependency tree information during the decoding time
  – Consider both subtree movements, and add them up to the translation model costs

• With cohesive constraints
  – The source dependency tree is available during the decoding time
  – Only consider either inside or outside movement, depending on the output of the interruption check algorithm
Outline

• Background of Reordering Models
• Source-side dependency tree reordering models
  – Modeling
  – Training
  – Decoding
• Experiments & Analysis
• Conclusion
Experiments setups

• **Baseline**: a phrase-based MT with lexicalized reordering model

• **Coh**: using cohesive constraints

• **DO / DOD / DOO**: using source-side dependency tree (SDT) reordering model with different parameter estimations

• **DO+Coh / DOD+Coh / DOO+Coh**: decoding with both SDT reordering model and cohesive constraints.
English-Spanish (Europarl)

- Source-side dependency tree reordering models and cohesive constraints obtained **improvements** over the lexicalized reordering models.
English-Iraqi (TransTac)

- Decoding with both source-side dependency tree reordering models and cohesive constraints often obtain the best performance.
Where are improvements coming from?
Test set breakdown

• Divide the test sets into three portions based on sentence-level TER of the baseline system

• $\mu$ and $\sigma$ are mean and standard deviation of the whole test set

• **Head, Tail** and **Mid** as the sentence whose score is lower than $\mu - \frac{1}{2} \sigma$, higher than $\mu + \frac{1}{2} \sigma$ and the rest
<table>
<thead>
<tr>
<th></th>
<th>june-08</th>
<th>nov-08</th>
<th>nc-test2007</th>
<th>news-test2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head</td>
<td>7.92</td>
<td>6.27</td>
<td>20.39</td>
<td>13.07</td>
</tr>
<tr>
<td>Mid</td>
<td>12.31</td>
<td>11.09</td>
<td>28.07</td>
<td>22.78</td>
</tr>
<tr>
<td>Tail</td>
<td>13.91</td>
<td>14.08</td>
<td>35.29</td>
<td>25.33</td>
</tr>
</tbody>
</table>
What is the most significant effect the source-tree reordering models contribute?
Numbers of Reorderings

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>1507</td>
<td>1684</td>
<td>39</td>
<td>24</td>
</tr>
<tr>
<td>Coh</td>
<td>2045</td>
<td><strong>2903</strong></td>
<td>46</td>
<td>21</td>
</tr>
<tr>
<td>DO</td>
<td><strong>2189</strong></td>
<td>2113</td>
<td>97</td>
<td>58</td>
</tr>
<tr>
<td>DO+Coh</td>
<td>1929</td>
<td>1900</td>
<td>155</td>
<td>88</td>
</tr>
<tr>
<td>DOD</td>
<td>1735</td>
<td>2592</td>
<td>123</td>
<td>60</td>
</tr>
<tr>
<td>DOD+Coh</td>
<td>2070</td>
<td>2021</td>
<td>148</td>
<td><strong>90</strong></td>
</tr>
<tr>
<td>DOO</td>
<td>1735</td>
<td>1785</td>
<td>164</td>
<td>49</td>
</tr>
<tr>
<td>DOO+Coh</td>
<td>1818</td>
<td>1959</td>
<td><strong>247</strong></td>
<td>66</td>
</tr>
</tbody>
</table>

- More reorderings can be generated without losing performance.

- The source-tree reordering models provide a more discriminative mechanism to estimate reordering events.

- Reordering is more language-specific than general translation models, and the conditions for a reordering event to happen vary among languages.
Outline

• Background & Motivations
• Source-side dependency tree reordering models
  – Modeling
  – Training
  – Decoding
• Experiments & Analysis
• Conclusions
Conclusions & Future Work

• Conclusions
  – Source-side dependency tree reordering models are helpful
    • Model reordering event with Inside/Outside subtree movements
  – The effectiveness was shown when comparing with a strong reordering model
  – Obtained improvements with 2 language pairs and also covered a training corpus sizes, ranging from 500K up to 1.3M sentence pairs

• Future work
  – Packed-forest dependency tree reordering models
Back up
A completed subtree

An open subtree

“c” is moving inside a subtree rooted at “b”

“d e” is moving outside a subtree rooted at “b”
What do you mean by introducing Inside/Outside notions?

• The movement of the **subtree inside** or **outside** a source subtree can be viewed as the decoder is leaving from the **previous** source state to the **current** source state.

• Tracking facts about the **subtree-to-subtree transitions** observed in the source side of word-aligned training data.
inside and outside probabilities for phrase “ask you”- “pedirle” according to three parameter estimation methods

<table>
<thead>
<tr>
<th></th>
<th>M_I</th>
<th>S_I</th>
<th>D_I</th>
<th>M_O</th>
<th>S_O</th>
<th>D_O</th>
</tr>
</thead>
<tbody>
<tr>
<td>DO</td>
<td>0.691</td>
<td>0.003</td>
<td>0.142</td>
<td>0.119</td>
<td>0.009</td>
<td>0.038</td>
</tr>
<tr>
<td>DOD</td>
<td>0.827</td>
<td>0.003</td>
<td>0.17</td>
<td>0.719</td>
<td>0.053</td>
<td>0.228</td>
</tr>
<tr>
<td>DOO</td>
<td>0.854</td>
<td>0.25</td>
<td>0.79</td>
<td>0.146</td>
<td>0.75</td>
<td>0.21</td>
</tr>
</tbody>
</table>
Distributions of Reordering Events

Observed **monotone & inside (M_I)** movements more often than other categories
Explicitly model phrase reordering distances

Distance-based (Och, 2002; Koehn et al., 2003)

Lexicalized phrase (Tillmann, 2004; Koehn et al., 2005; Al-Onaizan and Papineni, 2006)

Hierarchical phrase (Galley and Manning, 2008)

MaxEnt classifier (Zens and Ney, 2006; Xiong, et al., 2006; Chang, et al., 2009)

Put syntactic analysis of the target language into both modeling and decoding

Use source language syntax
Explicitly model phrase reordering distances

- Distance-based (Och, 2002; Koehn et.al., 2003)
- Lexicalized phrase (Tillmann, 2004; Koehn, et.al., 2005; Al-Onaizan and Papineni, 2006)
- Hierarchical phrase (Galley and Manning, 2008)
- MaxEnt classifier (Zens and Ney, 2006; Xiong, et.al., 2006; Chang, et.al., 2009)

Put syntactic analysis of the target language into both modeling and decoding

- Direct model target language constituents movement in either constituency trees (Yamada and Knight, 2001; Galley et.al., 2006; Zollmann et.al., 2008) or dependency trees (Quirk, et.al., 2005)
- Hierarchical phrase-based (Chiang, 2005; Shen et.al., 2008)

Use source language syntax