Dependency Forest for Statistical Machine Translation

Zhaopeng Tu\textsuperscript{1}, Yang Liu\textsuperscript{1}, Young-Sook Hwang\textsuperscript{2}, Qun Liu\textsuperscript{1}, and Shouxun Lin\textsuperscript{1}

\textsuperscript{1}Institute of Computing Technology \hspace{1cm} \textsuperscript{2}SKTelecom

\textsuperscript{1}\textit{Institute of Computing Technology} \hspace{1cm} \textsuperscript{2}\textit{SKTelecom}
Dependency Grammars in SMT

(string-to-dependency) (dependency-to-string) (dependency-to-dependency)

Bush held a talk with Sharon
bushi yu shalong juxing le huitan

(Shen et al., 2008; Xiong et al., 2007; Quirk et al., 2005)
String-to-Dependency Translation

(Shen et al., 2008)
String-to-Dependency Translation

(Chiang, 2007; Shen et al., 2008)
String-to-Dependency Translation

(Chiang, 2007; Shen et al., 2008)
String-to-Dependency Translation

(Shen et al., 2008)

Prob = \begin{align*}
P_T(saw) &= P(saw) \\
P_L(he | saw-as-head) &= P(he | saw-as-head) \\
P_L(a | boy-as-head) &= P(a | boy-as-head) \\
P_L(a | telescope-as-head) &= P(a | telescope-as-head) \\
P_R(boy | saw-as-head) &= P(boy | saw-as-head) \\
P_R(with | boy, saw-as-head) &= P(with | boy, saw-as-head) \\
P_R(telescope | with-as-head) &= P(telescope | with-as-head)
\end{align*}
Pipeline

source sentence -> GIZA -> word alignment

target sentence -> parser -> 1-best parse

dependency LM

rule set

decoder (Shen et al., 2008)
Pipeline

source sentence → GIZA → word alignment

target sentence → parser → 1-best parse

dependency LM

rule set

decoder (Shen et al., 2008)
Challenges

- string-to-dependency approach faces the major challenges:
  - vulnerable to parsing error
Solution

- forest provides an elegant solution to this problem (Mi and Huang, 2008)

\[
\begin{align*}
\text{NP} & \quad \text{VP} \\
\text{NR} & \quad \text{NN} & \quad \text{NR} & \quad \text{NN} & \quad = \\
\text{NP} & \quad \text{VP} & \quad \\
\text{NR} & \quad \text{NN}
\end{align*}
\]

from 1-best constituent tree to packed forest
from 1-best dependency tree to what?
saw

he

boy

boy

a

with

telescope

a
Dependency Forest

nodes

he\textsubscript{0,1} → saw\textsubscript{0,7} → boy\textsubscript{2,4} → boy\textsubscript{2,7} → with\textsubscript{4,7} → telescope\textsubscript{5,7} → a\textsubscript{5,6} → a\textsubscript{2,3}
Dependency Forest

\[
\text{he}_{0,1} \xrightarrow{e_1} \text{saw}_{0,7} \xrightarrow{e_2} \text{boy}_{2,4} \xrightarrow{a_2,3} \text{boy}_{2,7} \xrightarrow{a_5,6} \text{telescope}_{5,7} \xrightarrow{w_{4,7}} \text{with}_{4,7}
\]
source sentence

GIZA

word alignment

1-best parse

rule extractor

target sentence

parser

dependency

LM

tree-based

rule set

computer

decoder

(Shen et al., 2008)
Pipeline

source sentence

GIZA

word alignment

target sentence

parser

parse forest

dependency LM

rule set

rule extractor

decoder

(Shen et al., 2008)

forest-based
Forest-based Rule Extraction

he saw a boy with a telescope

ta kandao yige dai wangyuanjing de nanhai
Forest-based Rule Extraction

he_0,1
saw_0,7

boy_2,4 boy_2,7

with_4,7
telescope_5,7

he saw
a_2,3

a_5,6

he saw

he saw

ta kandao

yige dai wangyuanjing de nanhai
Forest-based Rule Extraction

saw

boy

a

with

a

boy

with

a

telescope

a
day
dai
ta

yige
dai

wangyuanjing
de

nanhai

saw

X₁

saw

X₁

kandao

X₁
Difficulty in Finding Phrase Pairs

too many choices
Difficulty in Finding Phrase Pairs

he saw a boy with a telescope

he saw a boy with a telescope
there are usually exponentially many well-formed structures over a target phrase in a dependency forest.
there are usually exponentially many well-formed structures over a target phrase in a dependency forest.
Difficulty in Finding Phrase Pairs

A well-formed structure is not necessarily a complete subtree, which GHKM algorithm (Galley et al., 2004) requires.
Forest-based Phrase Pair Extraction

bottom-up style

diagram:

- saw
- he
- boy
- with
- telescope
- a

2010/8/23
Forest-based Phrase Pair Extraction

bottom-up style

saw_{0,7}
he_{0,1}
boy_{2,4} boy_{2,7}
a_{2,3}
with_{4,7}
telescope_{5,7}
a_{5,6}
telescope
a
Forest-based Phrase Pair Extraction

bottom-up style

- saw
  - boy
  - he
  - a
  - telescope
- saw
  - boy
  - he
  - a
  - telescope
- bottom-up style
Forest-based Phrase Pair Extraction

bottom-up style

\[ \text{he}_{0,1} \xrightarrow{saw_{0,7}} \text{boy}_{2,7} \xrightarrow{a_{2,3}} \text{with}_{4,7} \xrightarrow{a_{5,6}} \text{telescope}_{5,7} \xrightarrow{a} \text{boy} \xrightarrow{saw} \]
Forest-based Phrase Pair Extraction

bottom-up style
Forest-based Dependency LM

saw_0,7
he_0,1
boy_2,4
boy_2,7
with_4,7
telescope_5,7
a_2,3
a_5,6

saw
he
boy
with

2010/8/23
COLING 2010, Beijing
Experiments
Setup

- Chinese-to-English translation
  - with a replication of string-to-dependency system (Shen et al., 2008)
- FBIS corpus
- 4-gram LM trained on Xinhua portion
- English-side parsed by the parser of Huang et al. (2009)
- 3-gram dependency LM trained on FBIS corpus plus 2M LDC corpus
## Tree-based Vs. Forest-based

<table>
<thead>
<tr>
<th>Rule</th>
<th>DepLM</th>
<th>NIST 04</th>
<th>NIST 05</th>
<th>NIST 06</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>tree</td>
<td>tree</td>
<td>33.97</td>
<td>30.21</td>
<td>30.73</td>
<td>19.6</td>
</tr>
<tr>
<td>tree</td>
<td>forest</td>
<td>34.42</td>
<td>31.06</td>
<td>31.37</td>
<td>24.1</td>
</tr>
<tr>
<td>forest</td>
<td>tree</td>
<td>34.60</td>
<td>31.16</td>
<td>31.45</td>
<td>21.7</td>
</tr>
<tr>
<td>forest</td>
<td>forest</td>
<td><strong>35.33</strong></td>
<td><strong>31.57</strong></td>
<td><strong>32.19</strong></td>
<td>28.5</td>
</tr>
</tbody>
</table>

- tree: tree-based
- forest: forest-based
- decoding time: seconds / sentence
## Tree-based Vs. Forest-based

<table>
<thead>
<tr>
<th>Rule</th>
<th>DepLM</th>
<th>NIST 04</th>
<th>NIST 05</th>
<th>NIST 06</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>tree</td>
<td>tree</td>
<td>33.97</td>
<td>30.21</td>
<td>30.73</td>
<td>19.6</td>
</tr>
<tr>
<td>tree</td>
<td>forest</td>
<td>34.42</td>
<td>31.06</td>
<td>31.37</td>
<td>24.1</td>
</tr>
<tr>
<td>forest</td>
<td>tree</td>
<td>34.60</td>
<td>31.16</td>
<td>31.45</td>
<td>21.7</td>
</tr>
<tr>
<td>forest</td>
<td>forest</td>
<td><strong>35.33</strong></td>
<td><strong>31.57</strong></td>
<td><strong>32.19</strong></td>
<td>28.5</td>
</tr>
</tbody>
</table>

tree: tree-based  
forest: forest-based  
decoding time: seconds / sentence
# Tree-based Vs. Forest-based

<table>
<thead>
<tr>
<th>Rule</th>
<th>DepLM</th>
<th>NIST 04</th>
<th>NIST 05</th>
<th>NIST 06</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>tree</td>
<td>tree</td>
<td>33.97</td>
<td>30.21</td>
<td>30.73</td>
<td>19.6</td>
</tr>
<tr>
<td>tree</td>
<td>forest</td>
<td>34.42</td>
<td>31.06</td>
<td>31.37</td>
<td>24.1</td>
</tr>
<tr>
<td>forest</td>
<td>tree</td>
<td>34.60</td>
<td>31.16</td>
<td>31.45</td>
<td>21.7</td>
</tr>
<tr>
<td>forest</td>
<td>forest</td>
<td>35.33</td>
<td>31.57</td>
<td>32.19</td>
<td>28.5</td>
</tr>
</tbody>
</table>

tree: tree-based
forest: forest-based
decoding time: seconds / sentence
## Rule Size

<table>
<thead>
<tr>
<th>Rule</th>
<th>Size</th>
<th>New Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>tree</td>
<td>7.2M</td>
<td>-</td>
</tr>
<tr>
<td>forest</td>
<td>7.6M</td>
<td>16.86%</td>
</tr>
</tbody>
</table>

new rules denotes rules extracted from non 1-best parses being used in 1-best derivations.
Conclusion and Future Work

- dependency forest provides an elegant solution to the problem of parsing error propagation
- very simple idea, but works very well in practice
  - ~1.4 BLEU points better than 1-best trees
- applicable to all dependency-based systems
  - other dependency-based systems (Quirk et al., 2005)
Forest offers more alternatives.

Thank you!

Thanks to Wenbin Jiang, and the anonymous reviewers.
Assign Probability to Hyperedge

he $\rightarrow$ saw $\rightarrow$ boy

he $\rightarrow$ saw $\rightarrow$ boy

$he_{0,1}$ $\rightarrow$ saw$_{0,7}$ $\rightarrow$ boy$_{2,4}$ $\rightarrow$ boy$_{2,7}$ $\rightarrow$ with$_{4,7}$
Assign Probability to Hyperedge

\[
c(e_1) = \exp \left( \frac{(13+22-1)}{3} \right)
\]

\[
c(e_2) = \exp \left( \frac{(13+22)}{2} \right)
\]
Assign Probability to Hyperedge

\[ c(e_1) = \exp \left[ \frac{(13+22-1)}{3} \right] \]
\[ c(e_2) = \exp \left[ \frac{(13+22)}{2} \right] \]

\[ p(e_1) = \frac{c(e_1)}{c(e_1) + c(e_2)} \]
\[ p(e_2) = \frac{c(e_2)}{c(e_1) + c(e_2)} \]