Jane: A Guide to RWTH’s Hierarchical Machine Translation Toolkit

Daniel Stein, David Vilar, Stephan Peitz and Hermann Ney
jane@i6.informatik.rwth-aachen.de
http://www.hltpr.rwth-aachen.de/jane

Sep 2010

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6
Computer Science Department
RWTH Aachen University, Germany
1 Introduction

► Hierarchical phrase-based translation toolkit, including
  ▶ Phrase extraction
  ▶ Decoding
  ▶ MERT training
► Toolkit written in C++, with tools in Python and Bash/Zsh
► Focus on efficiency and flexibility
► Parallelized operation under the Sun Grid Engine
► Extensions include syntax augmented models, advanced lexicon models, MIRA, …
► Jane is open-source for non-commercial purposes
► http://www.hltpr.rwth-aachen.de/jane
Outline

1 Introduction
2 Hierarchical Phrases
3 Extraction
4 Translation
5 Conclusions
Die Kommission schlägt vor, die Fristen zu verkürzen, und ich stimme dieser Forderung zu.

The Commission suggests shorter deadlines. I agree with this request.
<table>
<thead>
<tr>
<th>Request</th>
<th>This</th>
<th>With</th>
<th>Agree</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>ich</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stimme</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dieser</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forderung</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>zu</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I agree with this request.
<table>
<thead>
<tr>
<th></th>
<th>request</th>
<th>this</th>
<th>with</th>
<th>agree</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>ich</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stimme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dieser</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forderung</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>zu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Illustration

D. Stein D. Vilar S. Peitz H. Ney  A guide to Jane  Sep 2010
<table>
<thead>
<tr>
<th>request</th>
<th>this</th>
<th>with</th>
<th>agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>ich</td>
<td>stimme</td>
<td>dieser</td>
<td>Forderung</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X^~1</th>
</tr>
</thead>
<tbody>
<tr>
<td>X^~1</td>
</tr>
</tbody>
</table>

agree

I

I

D. Stein D. Vilar S. Peitz H. Ney  A guide to Jane  Sep 2010
Hierarchical Phrases

- Formalization as a synchronous CFG
- Rules of the form $X \rightarrow \langle \gamma, \alpha, \sim \rangle$, where:
  - $X$ is a non-terminal
  - $\gamma$ and $\alpha$ are strings of terminals and non-terminals
  - $\sim$ is a one-to-one correspondence between the non-terminals of $\alpha$ and $\gamma$

- Example:
  
  $X \rightarrow \langle \text{Ich stimme } X^1 \text{ zu, I agree with } X^1 \rangle$
  $X \rightarrow \langle \text{weil andere } X^1 \text{ nicht } X^2, \text{ because others have not } X^2 \text{ } X^1 \rangle$

- Additionally: Glue rules

  $S \rightarrow \langle S^1 X^2, S^1 X^2 \rangle$
  $S \rightarrow \langle X^1, X^1 \rangle$
3 Extraction

- Parallelized extraction and normalization of counts

- 2-pass extraction for filtering the target marginals
Additional Models

- Modular implementation of additional features
- Example usage (config file):

```
source=f.gz
target=e.gz
alignment=Alignment.gz
filter=devAndTest

additionalModels="syntax,parsematch"
extractOpts="--syntax.targetParsefile target.tree \ 
--parsematch.sourceParseFile source.tree \ 
--parsematch.targetParseFile targetTree"
```
DIY: Additional Models

► Inherit from:

AdditionalExtractionInformationCreator  Produces instances of AdditionalExtractionInformation

Main functions:

newSentence  Notifies of a new sentence pair
processCount  Called when a new phrase(-count) is created

AdditionalExtractionInformation  Wrapper class for the additional information required for the feature

Main functions:

add  Combines two instances of the class
    (e.g. the same feature is extracted from two different sentence pairs)
writePlain  For writing the information to disk
writePlainFinal  For writing the normalized score (if needed)

► Add your model to AdditionalExtractionInformationFactory.cc
Additional Models

Already implemented
► Soft syntactic labels [Venugopal & Zollmann$^+$ 09]
► Dependency information [Shen & Xu$^+$ 08]
► Parsematch information [Vilar & Stein$^+$ 08]
► Heuristic extraction features (non-aligned words, single word phrases, etc.)
► Alignment information
4 Translation

- Three running modes:
  - Single best translation
  - $n$-best translation
  - Server mode

- Cube pruning and cube growing

- On-demand loading of phrases for reduced memory footprint

- Four LM formats
  - Arpa
  - SRI binary format
  - RandLM
  - In-house binary format with on-demand loading

- Arbitrary number of LMs in search

- Sentence-level parallelization (Sun Grid Engine)
Translation: Principles

- Two passes: parsing and LM computation
- Parsing
  - CYK+ algorithm
  - Generation of an hypergraph
  - No LM scores are taken into account (directly)
  - Translations only implicitly computed

$$S \rightarrow AB$$
$$A \rightarrow u$$
$$B \rightarrow vw$$
Translation: Principles

► LM computation
  ▶ Traverse the hypergraph and compute $n$-best lists of derivations
    ○ Fixed size: cube pruning
    ○ On demand: cube growing
Cube Pruning

Translation Hyps $X_1$

Rules

0.3 0.8 1.1 1.6 2.1

0.2

0.9

1.3

2.3
Cube Pruning

Translation Hyps $X_1$

Translation Hyps $X_2$

Rules

0.3 0.8 1.1 1.6 2.1

0.2

0.8

0.3
Cube Pruning

Translation Hyps $X_1$

Translation Hyps $X_2$

Rules 0.3 2.3 2.9

0.3 0.8 1.1

0.2 0.8 1.3

0.9 1.5

1.3

2.3
Cube Pruning

Translation Hyps $X_1$

Translation Hyps $X_2$

Rules

0.3 2.3 2.9

0.3 0.8 1.1 1.6 2.1

0.2 0.8 1.3 1.6

0.9 1.5 2.0

1.3

2.3

2.9
Cube Pruning

Translation Hyps $X_1$

Translation Hyps $X_2$

Rules

0.3 2.3 2.9

0.3 0.8 1.1 1.6 2.1

0.2 0.8 1.3 1.6

0.9 1.5 2.0

1.3 1.9

2.3

D. Stein D. Vilar S. Peitz H. Ney A guide to Jane
Additional Models

- Modular implementation of additional features

- Example usage (config file):

```
[Jane]
decoder = cubePrune

[Jane.singleBest]
fileIn = f-dev
fileOut = f-dev.hyp

[Jane.CubePrune]
generationNbest = 100
secondaryModels = Syntax

[Jane.CubePrune.rules]
file = rules.bin

[Jane.CubePrune.LM]
file = lm.5gram.gz

[Jane.scalingFactors]
phraseS2T = 0.0391947693
phraseT2S = 0.0160933791
ibm1S2T = 0.0353934023
...  
LM = 0.0881110196
Syntax = 0.0023645551
syntaxPenalty = 0.0628653661
```
DIY: Additional Models

► Inherit from:

**SecondaryModel**  Computes additional scores

Main functions:

newSentence  Notifies of a new sentence to translate
scoreBackpointer  Compute scores for a derivation

► Add your model to SecondaryModelCreator.hh
Additional Models

Already implemented

- Extended lexicon models [Mauser & Hasan 09]
- Soft syntax labels [Venugopal & Zollmann 09]
- Dependency models [Stein & Peitz 10] based on [Shen & Xu 08]
- Reordering models [Vilar & Stein 10]
5 Conclusions

► Efficient toolkit for hierarchical phrase-based translation
► Easily extensible
► Parallelized operation
► Open source, free for non-commercial use
► http://www.hltp.rwth-aachen.de/jane
Jane: A Guide to RWTH’s Hierarchical Machine Translation Toolkit

Daniel Stein, David Vilar, Stephan Peitz and Hermann Ney
jane@i6.informatik.rwth-aachen.de
http://www.hltp.rwth-aachen.de/jane

Sep 2010

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6
Computer Science Department
RWTH Aachen University, Germany
References


