Hierarchical and Syntax Structured MT

Ashish Venugopal, Andreas Zollmann

InterACT, LTI, Carnegie Mellon University
Pittsburgh, PA
Stephan Vogel, Alex Waibel
1. Why pay the Syn - tax
2. Learning Syntax Augmented Grammars
3. Decoding with Syntax Augmented Grammars
4. Widening the S(A)MT pipeline
5. Tools and Conclusion
Surface form n-gram models are frustrating

- $P(\text{sweater}|\text{blue}) = \checkmark$
- $P(\text{sweater}|\text{red}) = ?$
- $P(\text{sweater}|\text{checkered}) = ?$

“Distortion” often distorts sentences

- Lexical / local distortion
- Models are too weak to effectively model translation equivalence
Typed Hierarchical Structure

- Model language as a hierarchical, typed process
- Prob. context free grammars rules are natural building blocks
- $VP \rightarrow ne \times 1 \text{ pas}, \text{ does not } VB_{x1}$
  - Example from “What’s in a translation rule” Galley et al.
Independence and Constraint

- VP → ne x1 pas, does not VB{x1}
- Translation of “ne ... pas” does not depend on words in VB
- Only (and any) VBs can be used in this structure
- Translate + Reorder
Probabilistic Synchronous Context Free Grammars

\[ X \rightarrow \langle \gamma, \alpha, \sim, w \rangle \]

- \( X \in N \) is a nonterminal
- \( \gamma \in (N \cup T_S)^* \) sequences of \( T_S, N \)
- \( \alpha \in (N \cup T_T)^* \) sequence of \( T_T, N \)
- \( \sim: \{1, \ldots, \#NT(\gamma)\} \rightarrow \{1, \ldots, \#NT(\alpha)\} \) is a one-to-one nonterminal mapping
- \( w \in [0, \infty) \) is a nonnegative real-valued weight assigned to the rule

- \( VP \rightarrow \) does not \( VB_{x1}, ne \ x1 \ pas \)
How do we translate?

- Bottom up chart parsing of source
- Source sequence $\rightarrow$ nonterminals and associated target translation
- Read translation from resulting parse tree
Decoding

- Initial source sentence

il ne va pas
Decoding

- VB → va, go

Il ne va pas → va, go
Decoding

\[ VP \rightarrow \text{does not } VB \]

\[ VP \rightarrow \text{ne } VB_{x1} \text{ pas, does not } x1 \]
Decoding

\[
S \rightarrow \text{he } VP
\]

\[
VP \rightarrow \text{does not } VB
\]

\[
VB \rightarrow \text{go}
\]

- S → il VP_{x1}, he x1
- Just one possible derivation!
Categories on Demand: Decoding vs Alignment Graph

S → he VP

VP → does not VB

VB → go

S

<table>
<thead>
<tr>
<th>NP</th>
<th>VP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRN</td>
<td>AUX</td>
</tr>
<tr>
<td>RB</td>
<td>VB</td>
</tr>
</tbody>
</table>

il ne va pas

il ne va pas
What kind of output do you want?

- If you want real trees · · ·
- Multilevel rules: Tree Substitution Grammars
- Non-contiguous units: Tree Insertion Grammars
  - Example from Chiang, Knight 2006
  - dat Jan Piet de kinderen zag helpen zwemmen
  - that John saw Peter help the children swim
- If you don’t care about trees · · ·
In the beginning there were · · ·

Target language parse trees
- “Syntax-Based”: **tree-driven**
  - Galley 2004, Galley et al. 2006, Marcu et al., 2006
  - Doesn’t respect bilingual phrases!

Phrase pairs, target language parse trees
- DOP-ish models: **tree-informed**
  - Extract rules from evidence (alignments, parse trees, *phrases*)
  - Chiang 2005, Zollmann 2006
  - Doesn’t respect target tree structure
In the beginning there were …

- Target language parse trees
  - “Syntax-Based” : **tree-driven**
    - Galley 2004, Galley et al. 2006, Marcu et al., 2006
    - Doesn’t respect bilingual phrases!

- Phrase pairs, target language parse trees
  - DOP-ish models : **tree-informed**
  - Extract rules from evidence (alignments, parse trees, *phrases*)
  - Chiang 2005, Zollmann 2006
  - Doesn’t respect target tree structure
Grammar Rule Extraction

- How can we learn probabilistic grammar rules?
- What do we learn them from?
  - French: *il ne va pas*
  - English: He does not go
  - Phrases (and their spans)
    - *il*, he does
    - *ne va pas*, does not go

- **Goal:** Annotate and Compose all initial rules
Alignment Graph

Why pay the Syn-tax

Learning Syntax Augmented Grammars
Decoding with Syntax Augmented Grammars
Widening the S(A)MT pipeline
Tools and Conclusion

Ashish Venugopal
MT Marathon, 04/17/07
Annotate and Compose

- For each phrase pair, assign a syntactic category based on the target words
- If we can’t find a category...
  - CCG style “slash” categories
  - Or ’X+Y’ and ’X+Y+Z’
  - Collect evidence from parse tree’s base
- Labels can come from anywhere!
- Compose multiple phrase pairs $\rightarrow$ complex rules.
$S \rightarrow \text{he does } RB + VB_{x1}, \ il \ x1$

The diagram illustrates the syntax tree for the sentence "he does not go," with the following parts:

- **S** (Sentence)
- **RB+VB** (Verb Phrase)
- **NP+Aux** (Noun Phrase and Auxiliary)
- **VP** (Verb Phrase)
- **NP** (Noun Phrase)
- **il** (French word)
- **ne** (French word)
- **va** (French word)
- **pas** (French word)
Why pay the Syn-tax
Learning Syntax Augmented Grammars
Decoding with Syntax Augmented Grammars
Widening the S(A)MT pipeline
Tools and Conclusion

Alignment Graph

INITIAL+ANNOTATED
- PRN → he, il
- VB → go, va
- VP → does not go, ne va pas
- S → he does not go, il ne va pas

GENERALIZE
- S → he VP_{x1}, il x1
- VP → does not VB_{x1}, ne x1 pas
- PRN+AUX → il, he does
Sample extracted rules

- \( S \rightarrow PRN_{x1} \text{ ne } VB_{x2} \text{ pas} , x1 \text{ does not } x2 \)
  - (handles ne pas construction)
- \( PRN+AUX \rightarrow PRN_{x1} , x1 \text{ does } \)
  - (adds an aux in English)
- \( S \rightarrow PRN + AUX_{x1} \ RB + VB_{x2} , x1 \times x2 \)
  - (facilitates nonlexical phrase for PRN+AUX)
- \( RB+VB \rightarrow \text{ ne va pas} , \text{ not go} \)
  - (fully lexicalized construction)
- \( S \rightarrow PRN + AUX_{x1} \text{ ne va pas} , x1 \text{ not go} \)
  - (facilitates use of PRN+AUX)
- \( RB+VB \rightarrow \text{ ne } VB_{x1} \text{ pas} , \text{ not } x1 \)
  - (alternative ne pas construction)
- \( S \rightarrow \text{ il ne va pas} , \text{ he does not go} \)
  - (whole sentence translation)
Decoding with Alternatives

- Initial source sentence

il ne va pas
Decoding with Alternatives

$$\text{VB} \rightarrow \{\text{go}, \text{goes}, \text{going}\} \mid \text{Cell} = 3$$

il ne va pas

- $\text{VB} \rightarrow \text{va, go}$
- $\text{VB} \rightarrow \text{va, goes}$
- $\text{VB} \rightarrow \text{va, going}$
Decoding

\[ VP \rightarrow \{\text{does not}\}, \{\text{no}\}, \text{VB}\{\text{go, goes, going}\} \mid \text{Cell} = 6 \]

\[
\begin{array}{c}
\text{VB} \\
\mid \text{go, goes, going} \\
\text{il ne va pas}
\end{array}
\]

- \( P(\text{go} | \text{does not}) \)
- \( P(\text{go} | \text{not}) \)
- \( \ldots \)
Decoding

\[ S \rightarrow \{\text{he, it}\} \ VP\{\cdots\} \mid \text{Cell} \mid = 12 \]

\[ \begin{align*}
\text{VP} & \rightarrow \{\text{does not}\}, \{\text{no}\} \ VB\{\text{go, goes, going}\} \\
\text{VB} & \rightarrow \{\text{go, goes, going}\} \\
\text{il} & \quad \text{ne} \quad \text{va} \quad \text{pas}\end{align*} \]

• Just one possible derivation (of rules)!
Integration of N-Gram Model

- Integrating N-Gram language model increases the virtual nonterminal space
- Theoretical Runtime: $\mathcal{O} \left( s^3 \left| \mathcal{N} \right| \left| \mathcal{T} \right|^{2(n-1)} \right)^K$
  - $K$: maximum number of NT pairs per rule
  - $s$: source sentence length.
  - $\mathcal{N}$: set of non-terminals
  - $\mathcal{T}$: set of terminals
  - $n$: order of n-gram LM
- $\mathcal{N} = 38K$ and $n = 3 + +$
Chart Structure

Each cell $i, j$ contains ⋯
- A set of target non-terminal categories $X_a, X_b$ ⋯
- Each target non-terminal contains equivalence classes ⋯
  - $\langle X_a, t_{\text{left}}, t_{\text{right}}, i, j \rangle_0$
  - Where each pair $t_{\text{left}}, t_{\text{right}}$ is unique
- Each equivalence class contains many chart items
Formation of a Chart Item

- Rule: \( X_S \rightarrow X_{np}^1 X_{pp}^2 X_{vp}^3 \leftrightarrow X_{np}^1 X_{vp}^3 X_{pp}^2 \)
- Example from Zhang et al.
- Terminal Productions: \( X_{np}^1 X_{pp}^2 X_{vp}^3 \)
  - \( \langle X_{pp}, [\text{with Sharon}], [\text{with Sharon}], i, j \rangle \)
  - \( \langle X_{pp}, [\text{in Sharon}], [\text{in Sharon}], i, j \rangle \)
  - \( \vdots \)
  - \( \langle X_{np}, [\text{held a}], [\text{a meeting}], i, j \rangle \)
  - \( \langle X_{np}, [\text{held-up a}], [\text{a meeting}], i, j \rangle \)
- Number of chart items formed: \( | X_{np} | \times | X_{pp} | \times | X_{vp} | \)
- We need to compute LM costs for each permutation
“If an item falls outside the beam, then any item generated using a lower...” ⋯

- Only generate the K-Best items of $|X_{np} | \times |X_{pp} | \times |X_{vp} |
  - Maintains an **ordered set** of equivalence classes
  - Better K-Best Extraction from Huang, Chiang 2005
  - Optimal K would be retrieved if not for the LM interaction

- **Pruning occurs across rules**
- **Prune away whole equivalence classes!**
Two Pass Decoding

- Two pass decoding:
  - Don’t increase virtual nonterminal space during 1st pass
  - Maintain un-explored chart item alternatives during 1st pass

- New Runtime: $O(s^3|\mathcal{N}|^K)$

- Search the resulting packed forest for new translations using a left-to-right heuristic search

- Venugopal, Zollmann, Vogel, NAACL 2007
  - Allows integration of flexible, high-order models
  - Limits LM calculations to successful decoding derivations
Decoding

\[ S \rightarrow \{he, it\} \ VP\{does\ not\ go\} \ |Cell| = 12 \]

- Only propagate 1 chart item per cell
- Keep the rest of them around for second stage search

\[ \text{il ne va pas} \]
Second Stage Search

\[ S \rightarrow \{ \text{he, it} \} \quad VP\{ \text{does not go} \} \mid \text{Cell} \mid = 12 \]

\[ VP \rightarrow \{ \text{does not} \}, \{ \text{no} \} \quad VB\{ \text{go, goes, going} \} \]

\[ VB \rightarrow \{ \text{go, goes, going} \} \]

- Only propagate 1 chart item per cell
- Keep the rest of them around for second stage search
- Results in a hypergraph of alternative sentence spanning parses
Why Left-to-Right Heuristic Search

- Left-to-right search allows integration of high-order LMs
- This is **better** than doing N-Best extraction and then re-scoring!
  - See Zollmann, Venugopal 2006 for improvements over re-scoring.
Left-to-Right Heuristic Search for N-Best Items

- Traverse the parse forest in Griebach-Normal Form
- Maintain a sentence spanning beam of trees

\[ X_{s0} \rightarrow X_{np}^1 X_{pp}^2 X_{vp}^3 \leftrightarrow X_{np0}^1 X_{vp0}^3 X_{pp0}^2 \]

- \( X_{s0} \ldots \leftrightarrow \text{Powell} \ X_{vp0}^3 X_{pp0}^2 \)
  - Used \( X_{np1}^1 \) update LM \( \mathcal{P}(\text{Powell}|\langle s \rangle) \)
- \( X_{s0} \ldots \leftrightarrow \text{Bowell} \ X_{vp0}^3 X_{pp0}^2 \)
  - Used \( X_{np2}^1 \): update LM \( \mathcal{P}(\text{Bowell}|\langle s \rangle) \)

\[ \vdots \]

- \( |X_{np}^1| \) items added to the beam
- Factor LM *in* to the real cost
- Factor *out* the words used in the estimate
- Update the LM estimate
Measuring Impact

- Two-stage search easily outperforms rescoring/naive pruning
- Cube Pruning vs Two-stage search
  - Evaluate LM cache misses vs Model Cost
  - Evaluate total time vs Model Cost
Experimental Results - Decoding

- IWSLT Evaluation - BTEC travel domain corpus
- 120K Parallel sentences, 1.2M target words
- Eval 500 sentences, average length 10.3 words
- Significance levels: approx 0.78 BLEU
Two Pass Decoding - LM Cache Misses

IWSLT - LM Cache Misses
Hierarchical

- CP
- H.Search

-3200
-3100
-3000
-2900
-2800
-2700
-2600
-2500
0.0E+00
2.5E+06
5.0E+06
7.5E+06

Number of LM Misses
Model Cost
CP
H.Search

0.174
0.175
0.177
0.180
0.181
0.182
0.186
0.188
0.191

IWSLT - LM Cache Misses Syntax

- CP
- H.Search

37400
37425
37450
37475
37500
2.0E+05
7.0E+05
1.2E+06

Number of LM Misses
Model Cost
CP
H.Search

0.205
0.206
0.207
0.207
0.207
0.206
0.2
SMT pipelines

- SMT systems are component driven
- SAMT: Alignments, Phrase Extraction, Parsing, Rule Extraction
- Each stage is considered as evidence for the next
What does it mean to be evidence?

- Each rule is associated with a feature vector
- Translation $\cong$ Parsing $\cong$ Finding best derivation of rules
- $p(D) = \frac{p_{LM}(tgt(D))^\lambda_{LM} \times \prod_{r \in D} \prod_i \phi_i(r)^{\lambda_i}}{Z(\lambda)}$
- $\lambda$ learned during MER - not during grammar induction
- $\phi$ contains MLE and binary/count style features
  - Target word count, IsSyntacticRule, IsBalanced rule etc.
What MLE style features do we use?

- \( \hat{p}(r|\text{lhs}(X)) \) : Probability of a rule given its l.h.s category
- \( \hat{p}(r|\text{src}(r)) \) : Probability of a rule given its source side
- \( \hat{p}(r|\text{tgt}(r)) \) : Probability of a rule given its target side
- \( \hat{p}(\text{ul}(\text{src}(r)), \text{ul}(\text{tgt}(r))|\text{ul}(\text{src}(r))) \) : Probability of the unlabeled source and target side of the rule given its unlabeled source side.
- \( \hat{p}(\text{ul}(\text{src}(r)), \text{ul}(\text{tgt}(r))|\text{ul}(\text{src}(r))) \) : Probability of the unlabeled source and target side of the rule given its unlabeled target side.

- Where do the counts come from?
Softening our notion of evidence

- Extracting a phrase doesn't mean it's correct!
- Extracting a rule with such a phrase is not correct either?
- What about syntactic categories?
  - Parse "errors" assign incorrect labels?
  - And propagate to incorrect rule arguments!
- We want a distribution over phrase composition, labeling decisions
Reflections on N-Best Lists and Parses

- A phrase from “buggy” alignments is buggy
- A phrase labeled from a “buggy” parse is buggy
- First best parses often contain errors
- Errors are usually the source of variance in n-best lists
Posterior models for MLE feature estimation

- N-Best alignments $a_1, \ldots, a_N$
- GIZA assigned probs $p(a_1 | e, f), \ldots, p(a_N | e, f)$ renormalized to $\hat{p}(a_i)$
- Same for parses $\hat{p}(\pi_j)$

$$cnt(r) = \sum_{i=1}^{N} \sum_{j=1}^{N'} \hat{p}(a_i) \cdot \hat{p}(\pi_j) \cdot \begin{cases} 1 & \text{if } r \text{ can be extracted from } e, f, a_i, \pi_j \\ 0 & \text{otherwise} \end{cases}$$

- Now use $cnt(r)$ in MLE estimates
- Exploit packed structural properties to correctly, efficiently calculate $cnt(r)$
Experimental Results

- IWSLT Evaluation - BTEC travel domain corpus
- GIZA trained to Model 4, Charniak parser 1000 best list
- Initial phrases based on Koehn 2003
- So far, only varied N for alignments vs parses separately
### Experimental Results - Lexicon from 1st best, Model 4

<table>
<thead>
<tr>
<th>$N, N'$</th>
<th>#Rules</th>
<th>#NTs</th>
<th>Dev</th>
<th>Test</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 1</td>
<td>300K</td>
<td>1771</td>
<td>23.7</td>
<td>19.8</td>
<td>1145</td>
</tr>
<tr>
<td>1, 1</td>
<td>311K</td>
<td>1781</td>
<td>23.7</td>
<td>21.2</td>
<td>1369</td>
</tr>
<tr>
<td>1..5, 1</td>
<td>490K</td>
<td>1894</td>
<td>24.3</td>
<td>21.0</td>
<td>2086</td>
</tr>
<tr>
<td>1..10, 1</td>
<td>582K</td>
<td>1947</td>
<td>24.3</td>
<td>20.1</td>
<td>2563</td>
</tr>
<tr>
<td>1..25, 1</td>
<td>747K</td>
<td>2026</td>
<td>24.4</td>
<td>20.1</td>
<td>3840</td>
</tr>
<tr>
<td>1..50, 1</td>
<td>911K</td>
<td>2072</td>
<td>24.8</td>
<td>21.1</td>
<td>5132</td>
</tr>
<tr>
<td>1..10, 1</td>
<td>1m</td>
<td>2212</td>
<td>26.0</td>
<td>22.2</td>
<td>13,406</td>
</tr>
<tr>
<td>1, 1..5</td>
<td>616K</td>
<td>2393</td>
<td>23.9</td>
<td>20.0</td>
<td>4291</td>
</tr>
<tr>
<td>1, 1..10</td>
<td>850K</td>
<td>2633</td>
<td>24.0</td>
<td>20.1</td>
<td>7237</td>
</tr>
<tr>
<td>1, 1..10</td>
<td>652K</td>
<td>2407</td>
<td>25.9</td>
<td>X</td>
<td>13,396</td>
</tr>
</tbody>
</table>

**Table:** Grammar statistics and translation quality (IBM-BLEU) on development and test set and when integrating $N$-best alignments and $N'$-best parses. Decoding time in seconds is on all 500 sentences.
Some interesting rules

- Rules that weren’t found in the 1-best list
- IWSLT has non-punctuated source, punctuated targets

<table>
<thead>
<tr>
<th>count</th>
<th>source</th>
<th>target</th>
<th>LHS</th>
<th>NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>247.93</td>
<td>请</td>
<td>please .</td>
<td>@UH+</td>
<td></td>
</tr>
<tr>
<td>210.69</td>
<td>请</td>
<td>please .</td>
<td>@VB+</td>
<td></td>
</tr>
<tr>
<td>162.06</td>
<td>想</td>
<td>'d</td>
<td>@MD</td>
<td></td>
</tr>
<tr>
<td>153.42</td>
<td>我</td>
<td>, I</td>
<td>@, +PRP</td>
<td></td>
</tr>
<tr>
<td>146.32</td>
<td>我</td>
<td>I have</td>
<td>@PRP+</td>
<td>AUX</td>
</tr>
<tr>
<td>141.96</td>
<td>我</td>
<td>.</td>
<td>@</td>
<td></td>
</tr>
<tr>
<td>141.75</td>
<td>的</td>
<td>in</td>
<td>@IN</td>
<td></td>
</tr>
</tbody>
</table>
System Output
System track record

- Beating or matching phrase based baselines
- Small and medium data tasks
- Chinese-English IWSLT
- (French/Spanish)-English Europarl
- Chinese-English NIST
IWSLT Chinese English

<table>
<thead>
<tr>
<th>Rules</th>
<th>Dev IBM-BLEU</th>
<th>Test IBM-BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>X grammar</td>
<td>21.25</td>
<td>18.08</td>
</tr>
<tr>
<td>Pharaoh</td>
<td>22.0</td>
<td>19.3</td>
</tr>
<tr>
<td>SAMT</td>
<td>23.50</td>
<td>20.04</td>
</tr>
</tbody>
</table>

Table: Comparison of translation-models system using “SmartCase”, evaluated on the official case and punctuation sensitive IBM-BLEU metric.
Spanish-English

- 2000 sentences Test 06 Spanish English Europarl
- PhraseBased: 31.76
- SyntaxAugmented: 32.15
- Minimal impact of Re-ordering for Spanish
  - Development data (tuned)
  - Window 1: 31.98
  - Window 2: 32.24
  - Window 3: 32.30
  - Window 4: 32.26
  - Syntax: 32.48
Chinese-English NIST

- Chinese-English NIST Evaluation - 1 day worth of training time - 3-gram LM on target side of data
- Case Sensitive Official NISTBLEU
- No. Rules applicable to Dev and Test.
  - X: Style of Chiang 2005
  - Penn: Retains only those that are constituents
  - CCG+: Assigns categories to almost all lexical phrases

<table>
<thead>
<tr>
<th>Grammar</th>
<th>NTs</th>
<th>Rules</th>
<th>Time</th>
<th>Dev (MT03)</th>
<th>Test (MT05)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>2</td>
<td>197K</td>
<td>1.9h</td>
<td>23.5</td>
<td>X</td>
</tr>
<tr>
<td>Penn</td>
<td>73</td>
<td>191K</td>
<td>0.3h</td>
<td>22.8</td>
<td>21.1</td>
</tr>
<tr>
<td>CCG+</td>
<td>38,861</td>
<td>795K</td>
<td>0.9h</td>
<td>28.7</td>
<td>26.2</td>
</tr>
</tbody>
</table>
Open Source Tools

- All tools available at www.cs.cmu.edu/~zollmann/samt/
- `extractrules.pl` - identify Syn CFG rules
- `filterrules.pl` - score and prune rules
- `FastTranslateChart` - Chart parser decoder, N-best lists, MER
- `MER` - standalone MER toolkit