Syntax Based Machine Translation

Fabienne Braune

University of Stuttgart
Institute for Natural Language Processing
D 70174 Stuttgart
braunefe@ims.uni-stuttgart.de
Outline

- Overview of syntax-based MT using SCFG
- Motivation for more powerful tree formalisms (STSG)
- Top Down Tree Transducers (TOP)
- Extended Top Down Tree Transducers (XTOP)
- STSG and XTOP
Syntax-Based SMT

- Machine Translation System:
  - Needs: set of weighted rules: **training**
  - Gives: k-best derivation given the input parse tree: **decoding**
  - "Best derivation", "Best rule": **model**
Input and Model of a syntax-based SMT system

- Input (source language): *The red car ran fast*

- Parsed:

```
  S
    NP
      DET the
      JJ red
      NNS car
    VP
      VBP ran
      ADVP fast
```

- Model:

\[
\text{argmax}(\text{score}(\text{tree}, e, f)), \ \text{where} \\
\text{score}(\text{tree}, e, f) = \prod_i \text{RULE}_i
\]
Set of Weighted Rules : SCFG rules

1. $< S, S > \xrightarrow{1.0} < NP_1 \ VP_2, NP_1 \ VP_2 >$
2. $< NP, NP > \xrightarrow{0.6} < DET_1 \ JJ_2 \ NNS_3, DET_1 \ NNS_3 \ JJ_2 >$
3. $< NP, NP > \xrightarrow{0.4} < \text{The JJ}_1 \ car, \ La \ voiture \ JJ_1 >$
4. $< VP, VP > \xrightarrow{1.0} < VBP_1 \ ADVP_2, VBP_1 \ ADVP_2 >$
5. $< ADVP, ADVP > \xrightarrow{1.0} < RB_1, RB_1 >$
6. $< VBP, VBP > \xrightarrow{0.7} \text{ran, roulait} \quad 10. \ < VBP, VBP > \xrightarrow{0.3} \text{ran, roulat}$
7. $< JJ, JJ > \xrightarrow{1.0} \text{red, rouge} \quad 11. \ < RB, RB > \xrightarrow{0.6} \text{fast, vite}$
8. $< \text{Det, Det} > \xrightarrow{0.5} \text{The, La} \quad 12. \ < \text{NNS, NNS} > \xrightarrow{1.0} \text{car, voiture}$
9. $< \text{Det, Det} > \xrightarrow{0.5} \text{The, Le} \quad 13. \ < RB, RB > \xrightarrow{0.4} \text{fast, rapidement}$
Find best derivation

- Best derivation: \( \text{argmax}(\text{score}(\text{tree}, e, f)) \)

- Apply rules on input parse tree bottom-up:
  - Add fully lexicalized rules
  - Add partly lexicalized rules
  - Add non lexicalized rules

- Generate \( k \) parse trees in target language
- Take the best one: multiply rule weights in derivation
- Take yield: \textit{La voiture rouge roulait vite}
Exercise 1: Find best derivation

The red car ran fast
Need for more powerful models: STSG

- \textit{John misses Mary} $\rightarrow$ \textit{Marie manque à Jean}

- Solution: use tree fragments instead of flat trees

- Can be modelled by Extended Top Down Tree Transducers
Top Down Tree Transducers: Syntax

- $M = (Q, \Sigma, \Delta, I, R)$ where:
  - $Q$ is a finite set of states,
  - $\Sigma$ and $\Delta$ are alphabets of input and output symbols,
  - $I \subseteq Q$ is a set of initial states,
  - $R \subseteq Q(\Sigma(X)) \times T_\Delta(Q(X))$ is a finite set of (rewrite) rules
Example TOP I

\[\rho_1 \quad \text{plus} \quad \rightarrow \quad \text{plus} \quad d \quad d\]

\[x_1 \quad x_2 \quad x_1 \quad x_2\]

\[Q = \{d\}, \Sigma = \{\text{plus}\}, \Delta = \{\text{plus}\}, l = \{d\}\]
Example TOP II

\[ \rho_2^{\text{mult}} x_1 x_2 \rightarrow \text{plus} \]

\[ Q = \{ d, i \}, \Sigma = \{ \text{mult} \}, \Delta = \{ \text{plus, mult} \}, I = \{ d \} \]
Example TOP III

\[ \rho_3 \sin x_0 \rightarrow \text{mult} \]

\[ Q = \{ d, i \}, \Sigma = \{ \sin, a \}, \Delta = \{ \text{mult}, \cos \}, l = \{ d \} \]
Example TOP IV

\[ \rho_4 \xrightarrow{i} a \]
\[ \rho_5 \xrightarrow{d} \]

\[ Q = \{d, i\}, \Sigma = \{a\}, \Delta = \{a, b\}, l = \{d, i\} \]
Example TOP : Summary

\[ Q = \{d, i\}, \Sigma = \{\text{plus}, \text{mult}, \text{sin}, a\}, \Delta = \{\text{plus}, \text{mult}, \text{cos}, \text{sin}, a\}, l = \{d\} \]
Top Down Tree Transducer: Semantics

- Tree Transformation:

$$\tau_M = \{(t, u) \in T_\Sigma \times T_\Delta \mid \exists q \in I: q(t) \Rightarrow_M^* u\}$$

where $$\Rightarrow_M^*$$ is the reflexive, transitive closure of $$\Rightarrow_M$$. 

![Diagram](image.png)
• Compute derivatives:
  • infix notation: \((f + g)' = f' + g'\)
  • prefix notation: \(deriv(plus(f, g)) = plus(deriv(f), deriv(g))\)

  • infix notation: \((fg)' = f'g + fg'\)
  • prefix notation:
    \(deriv(mult(f, g)) = plus(mult(f, deriv(g)), mult(g, deriv(f)))\)

  • ...
Top Down Tree Transducer: Derivation I

\[
d \cdot \sin a \cdot \mult a a \Rightarrow \rho_1
\]
Top Down Tree Transducer: Derivation III

\[
\text{plus} \quad \text{mult} \quad d \quad \Rightarrow \quad \rho_2
\]

\[
\text{cos} \quad d \quad \text{mult} \\
i \quad a \quad a \quad a
\]

\[
\text{plus} \quad \text{mult} \quad \text{mult} \\
i \quad a \quad d \quad i \quad d \quad i \\
i \quad a \quad a \quad a \quad a
\]
Top Down Tree Transducer: Derivation IV

\[
\text{mult} \quad \text{plus} \quad \text{mult} \quad \text{mult} \\
\text{cos} \quad d \quad \text{mult} \quad \text{mult} \\
i \quad a \quad d \quad i \quad d \quad i \\
a \quad a \quad a \quad a \quad a
\]
\[
\Rightarrow \rho_{4,5}
\]

\[
\text{mult} \quad \text{plus} \quad\text{mult} \quad \text{mult} \\
\text{cos} \quad b \quad \text{mult} \quad \text{mult} \\
a \quad b \quad a \quad b \quad a
\]
Top Down Tree Transducer: Derivation Summary

\[
\begin{align*}
\text{plus} & \quad \Rightarrow \rho_1 \\
\sin & \quad \text{mult} & \quad \Rightarrow \rho_3 \\
a & \quad a & \quad \cos & \quad \text{mult} \\
\text{plus} & \quad \Rightarrow \rho_2 \\
\text{mult} & \quad \Rightarrow \rho_{4,5} \\
\cos & \quad \text{mult} & \quad \text{plus} \\
i & \quad a & \quad b & \quad \text{mult} & \quad \text{mult} \\
a & \quad a & \quad a & \quad a & \quad a & \quad a
\end{align*}
\]
Exercise 2: TOP derivation

\[
\text{mult} \\
\text{sin} \quad \text{sin} \\
\quad a \quad a
\]
When TOP fails : extend to XTOP

- Translation of verbal complex :
  The Serbs **have completed** the negotiations
  Die Serben **haben** die Verhandlungen **beendet**

```
C
  NP
    VP
      VAUX
      VPART

C'
  NP
    VP
      VAUX
      NP
      VPART
```

- We want to reverse VPART and NP
- But : we have to consume C and VP
  ⇒ Allow to consume $T_\Sigma$ insted of single $\sigma \in \Sigma$
Extended Top Down Tree Transducers: Syntax

- \( M = (Q, \Sigma, \Delta, I, R) \) where:
  - \( Q \) is a finite set of states,
  - \( \Sigma \) and \( \Delta \) are alphabets of input and output symbols,
  - \( I \subseteq Q \) is a set of initial states,
  - \( R \subseteq Q(T_\Sigma(X)) \times T_\Delta(Q(X)) \) is a finite set of (rewrite) rules

\[
\begin{array}{c}
\rho_1 \\
q_C \\
C \\
\text{VP} \\
x_1 \\
x_2 \quad x_3 \quad x_4 \\
q_{NP} \\
q_{VP} \\
C' \\
\text{VP} \\
x_1 \\
x_2 \quad x_4 \quad x_3 \\
q_{VA} \\
q_{NP} \\
q_{VP}
\end{array}
\]
Extended Top Down Tree Transducer : Semantics

- Tree Transformation:

\[ \tau_M = \{(t, u) \in T_\Sigma \times T_\Delta \mid \exists q \in I : q(t) \Rightarrow^*_M u\} \]
• *John misses Mary* → *Marie manque à Jean*

• Solution: use tree fragments instead of flat trees

• Can be modelled by Extended Top Down Tree Transducers
XTOP in syntax-based MT II

• Input Tree :

\[
S \\
\quad NP \quad VP \\
\quad \quad John \quad V \quad NP \\
\quad \quad \quad misses \quad Mary
\]

• XTOP Rules :

\[
\rho_1 x_1 \rightarrow q NP \\
\rho_2 x_2 \rightarrow VP \\
\rho_3 x_1 \rightarrow NP
\]

\[
S \\
q NP \\
\quad \quad VP \\
\quad \quad \quad V \\
\quad \quad \quad \quad \quad \quad misses \\
\quad \quad \quad \quad \quad \quad \quad x_2
\]

\[
S \\
q NP \\
\quad VP \\
\quad \quad V \\
\quad \quad \quad \quad \quad \quad manque a \\
\quad \quad \quad \quad \quad \quad \quad \quad x_1
\]

\[
Pp_2 \rightarrow Jean \\
q NP \rightarrow NP \\
\rho_3 \rightarrow Marie
\]
• Derivations:
  ⇒ Let’s do one together...
Questions

- Thank you for your attention
- Questions ?
Topics I : SCFS and Top Down Tree Transducers

• Chiang. 2007. *Hierarchical phrase-based translation*
  
  • What are Hierarchical Phrases ? What are SCFG ?
  • How is the model defined ?
  • How do we extract and train Hierarchical Phrases ?
  • How does decoding work ?
  • How is the system evaluated ?

• Knight and Graehl. 2005. *An overview of probabilistic tree transducers for natural language processing*
  
  • Why do we want to use tree-based devices models in MT ?
  • What are top down tree transducers and how do they work ?
  • What are their properties. Why are they relevant to MT ?
  • Sketch a translation system using top-down tree transducers (trained rules are given).
Topics II : STSG and STAG

  - What are Synchronous Tree Substitution Grammars?
  - How are STSG extended to probabilistic STSG?
  - Present the proposed Tree Parsing algorithm?
  - How is this algorithm used in training and decoding?

  - What are Synchronous Tree Adjoining Grammars?
  - Explain the relation/difference to STSG
Topics IV: Training

  - How do we distinguish good derivations from bad ones?
  - How do we extract rules from derivations?
  - Present the proposed rule extraction algorithm
  - How is the obtained set of rules evaluated?

  - What are Regular Tree Grammars? Extended Top Down Tree Transducers (xR-transducers)?
  - How do we generate derivation trees for xR-transducers?
  - How are inside/outside weights defined? How are they used in EM training?
  - How do we extend the training procedure to the tree-to-string case?
Topics V : All Together

• Maletti 2010. *A tree transducer model for synchronous tree-adjoining grammars*
  
  • What are Extended Top Down Tree Transducers (XTOP)? Relation to STSG?
  • What is the relation/difference between STSG and STAG? XTOP and STAG?
  • Why is the shown result interesting for practical applications?
  • What are embedded Tree Transducers?
  • Present the proposed Tree Transducer Model?

• DeNeefe and Knight. 2009. *Synchronous tree adjoining machine translation*
  
  • Explain the proposed model
  • What training procedure is used?
  • How is decoding performed?
  • How is the system evaluated?