Machine Translation Topics

Fabienne Braune

University of Stuttgart
Institute for Natural Language Processing
– Computational Linguistics –
Azenbergstr. 12
D 70174 Stuttgart
braunefe@ims.uni-stuttgart.de
Syntax-Based SMT

- **Machine Translation System**:
  - Needs: set of weighted rules: \textit{training}
  - Gives: k-best derivation given the input parse tree: \textit{decoding}
  - ”Best derivation”, ”Best rule”: \textit{model}
Example 1 : SCFG

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

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

- Set of weighted rules : start $S \rightarrow < S_1, S_1 >$

\[
S \xrightarrow{1.0} < NP_1 VP_2, NP_1 VP_2 > \\
NP \xrightarrow{1.0} < DET_1 JJ_2 NNS_3, DET_1 NNS_3 JJ_2 > \\
VP \xrightarrow{1.0} < VBP_1 ADVP_2, VBP_1 ADVP_2 > \\
VBP \xrightarrow{0.7} ran, roulait \\
JJ \xrightarrow{1.0} red, rouge \\
ADVP \xrightarrow{1.0} < RB_1, RB_1 > \\
Det \xrightarrow{1.0} The, La \\
NNS \xrightarrow{1.0} car, voiture \\
VBP \xrightarrow{0.3} ran, roulait \\
RB \xrightarrow{1.0} fast, vite
\]
Example 1: Derivation

- Apply rules on input parse tree
- Generate k parse trees in target language
- Take the best one: multiply rule weights in derivation
- Take yield: *La voiture rouge roulait vite*

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

```
S
  NP
    DET la
    NNS voiture
    JJ rouge
  VP
    VBP roulait/roulât
    ADVP vite
```
Where SCFG fail

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

- Solution: use tree fragments instead of flat trees

- Formal Models: Synchronous Tree Substitution Grammars / Extended Top Down Tree Transducers

- Fail when dealing with discontinuous constituents
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

• Galley, Hopkins, Knight and Daniel Marcu. 2004. *What’s in a translation rule?*
  
  • 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?