

CMU-UKA Syntax Augmented Machine Translation

Ashish Venugopal, Andreas Zollmann, Stephan Vogel, Alex Waibel

InterACT, LTI, Carnegie Mellon University
Pittsburgh, PA



Outline

- 1 Model of Translation
- 2 Decoding Strategy
- 3 Data Processing
- 4 Evaluation

Issues Addressed

- Extended Translation models for SMT
- Data Processing (Case + Punctuation)
- Syntax Augmented Machine Translation via Chart Parsing

Generalized Rules

- Extending the model of translational equivalence
- he does not go , *il ne va pas*
- does not go , *ne va pas*
- does not X , *ne X pas*
 - Chiang, 05, Watanabe et al. 06
- S → does not VB0 , *ne X0 pas*
 - Galley, 04, Zollmann, Venugopal, 06, Galley et al. 06

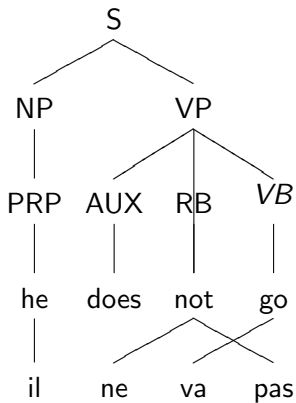
Automatic Induction of Rules

- Start with word alignments (a) on f, e + phrase based model
- We want to model **target language** syntactic structure
- Take advantage of target side parse tree π

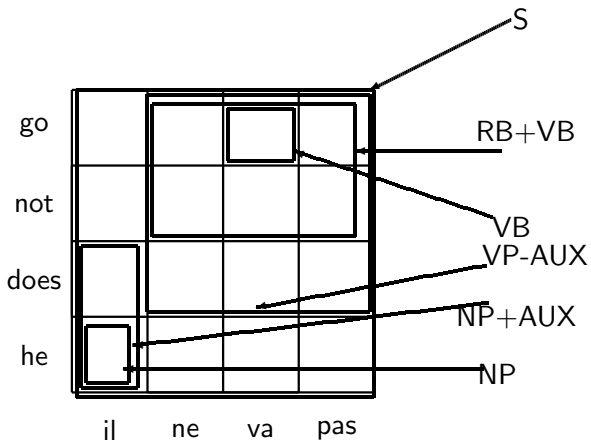
Data Oriented Approach

- Consider $\pi, f, e, phrases(a)$ as given
- Extract rules consistent with $phrases(a)$
- Begin with contiguous src, tgt phrases
- Annotate, Generalize, Re-order
- Producing “flattened” xRS style rules
- CCG style operations-add/subtract

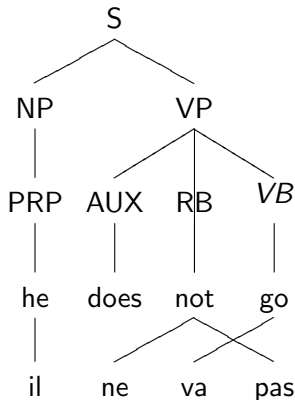
Alignment Graph



Extraction Approach



Alignment Graph



- INITIAL+ANNOTATED

- PRP → he, *il*
- VB → go, *va*
- VP → does not go, *ne va pas*
- S → he does not go, *il ne va pas*
- PRP+AUX → *il, he does*

- GENERALIZED

- S → he VP_0 , $il\ x_0$
- VP → does not VP_0 , $ne\ x_0\ pas$

All rules extracted

- $S \rightarrow PP \text{ ne VB pas}, 1 \text{ do not } 2$
- $PP+AUX \rightarrow PP, 1 \text{ do}$
- $S \rightarrow PP+AUX \text{ RB+VB}, 1 \text{ } 2$
- $RB+VB \rightarrow \text{ne vais pas}, \text{not go}$
- $S \rightarrow PP+AUX \text{ ne vais pas}, 1 \text{ not go}$
- $S \rightarrow \text{je RB+VB}, i \text{ do } 1$
- $S \rightarrow PP \text{ RB+VB}, 1 \text{ do } 2$
- $VP \rightarrow \text{ne VB pas}, \text{do not } 1$
- $S \rightarrow \text{je ne VB pas}, i \text{ do not } 1$
- $S \rightarrow \text{je VP}, i \text{ } 1$
- $PP \rightarrow \text{je}, i$
- $RB+VB \rightarrow \text{ne VB pas}, \text{not } 1$
- $VB \rightarrow \text{vais}, \text{go}$
- $S \rightarrow \text{je ne vais pas}, i \text{ do not go}$

Chart Parser Based Decoding

- Earley style bottom-up parsing
- We do not require rule binarization as in (Huang, 06)
- Integrated N-gram Language Model (Wu, 98)
 - Single-pass Cube Pruning (Chiang, 05)
 - Multi-pass heuristic search (Zollmann, Venugopal, 06)
- All rules stored in Berkeley DB

Log-Linear translation model features

- Source, target conditioned lexical weights as in Koehn, 2003
- Relative frequencies conditioned on ...
 - Left-hand side category
 - Source phrase
 - Target phrase
- Counters: Rule applications, Target words generated
- Bias terms
 - IsPurelyLexical
 - IsPurelyAbstract
 - IsXRule (non-syntactic span)
 - IsGlueRule
- Penalty features
 - Rareness: $e^{1 - RuleFrequency}$
 - Lexical Length Balance: $|MeanTargetSourceRatio \times |src| \times tgt|$

Case Handling

- Training vs Translation time options
- TrainLowerCase: training lower case + true-case output
- TrainTrueCase: training true case
- SmartCase: the ambiguity is in the first word!
 - Upper-case words used within the sentence tend to be consistent
 - For each first-word - estimate non-first word case frequency
 - Use most common case consistently
 - Upper-case first word of output

SmartCasing on Training Data

English word	Frequency	Fr. of lower-case variant
I	21611	10
Japan	1939	0
Japanese	1816	0
Tokyo	813	0
Hotel	703	1166
Mr.	691	0
New	487	315
York	438	0
English	372	0
Boston	318	0

Punctuation

- Remove punctuation in source training to match test
- Move sentence-end marks “.!?” to the beginning
- Train model that generates punctuation
- Move leading punctuation to end in output

Times and Numbers

- Number handling based on quantifier markers
- Specific handling for Chinese “point” symbol

Data Conditions

- C-Star Data Track participants
- Focussing on correct transcription evaluation
- Note: Error in data alignment, only Supplied Data was parallel
- 2006 Development development data only

Comparison of Casing - PostEval

Processing	Dev IBM-BLEU	Test IBM-BLEU
TrainLowerCase	22.04 (23.91)	19.16 (21.55)
TrainTrueCase	22.47 (24.05)	19.23 (20.48)
<i>SmartCase</i>	23.50 (25.17)	20.04 (21.76)

Table: *Comparison of different case-handling methods using the syntax-augmented translation system evaluated on the official case- and punctuation-sensitive IBM-BLEU metric. The numbers in parentheses indicate the IBM-BLEU score when case (but not punctuation) is ignored.*

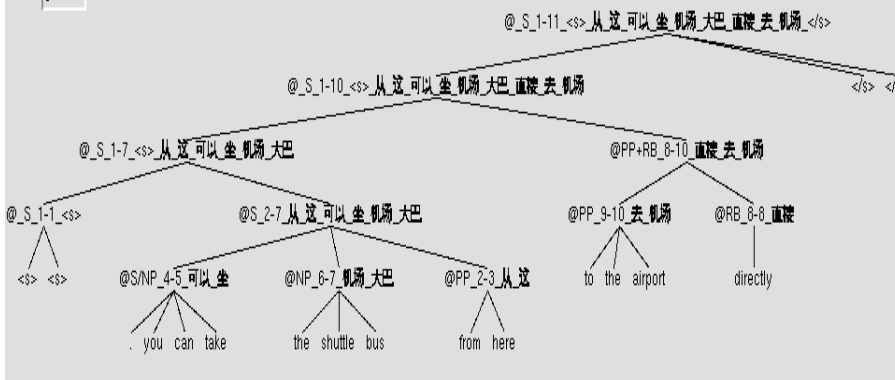
Comparison of Models - PostEval

Rules	Dev IBM-BLEU	Test IBM-BLEU
Chiang-sim	21.25	18.08
Pharaoh	23.2 → 22.0	19.3
SAMT	23.50	20.04

Table: *Comparison of translation-models system using “SmartCase”, evaluated on the official case and punctuation sensitive IBM-BLEU metric. Note the change for the Pharaoh result. The initial result was run with the incorrect version of BLEU*

Example

Item: 0



Decoding Time

- Decoding Time depends on the kind of rules extracted
- Two particularly problematic rules
 - Purely abstract - flattened full sentence structure
 - Target only - insertion of target words with no src
- Runtime in evaluation: 50 minutes for Dev 06
- Runtime w/o abstract/tgt only: 5 minutes for Dev 06
- End-to-end system available at www.cs.cmu.edu/~zollmann/samt

Future Work

- Shift emphasis to speech-translation task
- Evaluate parsing for ASR output
- Directly decode ASR lattices