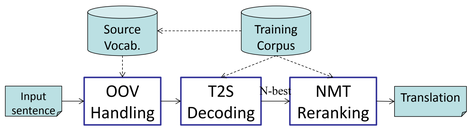
NAVER Machine Translation System for WAT 2015

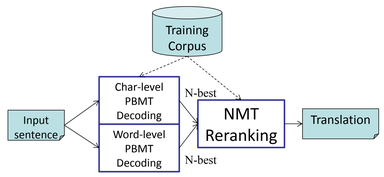
Hyoung-Gyu Lee[1], Jaesong Lee[1], Jun-Seok Kim[1] and Chang-Ki Lee[2]
[1]NAVER LABS, NAVER Corp. and [2]Kangwon National University

System Overview

English to Japanese (En-Ja)



Korean to Japanese (Ko-Ja)

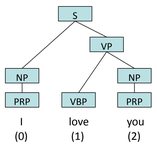


English to Japanese

❑ Language Analyzer
- ❑ **English**: Moses tokenizer & Berkeley parser
- ❑ **Japanese**: In-house tokenizer and POS tagger

❑ Tree-to-String Syntax-based SMT
- ❑ Use source parse trees
- ❑ Synchronous context free grammar (SCFG) rules
- ❑ Char parsing decoder with cube pruning

❑ Rule Augmentation
- ❑ Known as Syntax-Augmented Machine Translation (Zollmann and Venugopal, 2006)
- ❑ Extract more rules by modifying source parse tree
- ❑ Given a parse tree, produce additional nodes

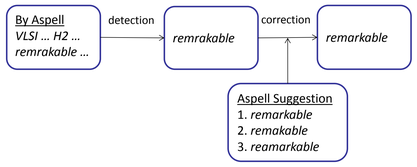


Baseline nodes	Additional nodes
0-0 PRP	1-2 VBP+PRP
0-0 NP	0-2 PRP+VP
1-1 VBP	0-1 PRP++VBP
2-2 PRP	
2-2 NP	
1-2 VP	
0-2 S	

❑ Handling OOV
(1) **Hyphen Word Split**
- ❑ e.g. *nano-laminate* → *nano laminate*

(2) **Spell Error Correction**
- ❑ Use open source spell checker, Aspell
- ❑ Detection based on skip rule
  - ❑ Capitals, numbers, symbols
- ❑ Correction based on edit distance
  - ❑ Large gap causes wrong correction
  - ❑ Select one with the shortest distance
    - ❑ among top-3 suggestion

❑ Example



Korean-to-Japanese

❑ Language Analyzer
- ❑ **Korean**: Mecab-ko w/o POS tagger
- ❑ **Japanese**: Juman w/o POS tagger

❑ Combination of Two PBMT
- ❑ Word-level PBMT
  - ❑ 5-gram LM, max-phrase-len=7
- ❑ Char-level PBMT
  - ❑ Char-level tokenization
  - ❑ 10-gram LM, max-phrase-len=10
- ❑ Methods for selecting the best hypothesis

(1) **Rule-based**:
- Char-level PB, if an input has an OOV
- Word-level PB, otherwise

(2) **NMT**:
- Rerank n-best lists of two PBMT

❑ Why Char-level PBMT
- ❑ Mis-tokenization problem in word-level PBMT
  → cause many OOVs

Ko/Ja representation of "Diisononyl phthalate":
- ❑ Ko: 디 / 이소노 / 닐 / 프탈 / 레이트 (5 tokens, 1 OOV)
- ❑ Ja: ジイソノニルフタレート (1 token)

- ❑ Both Ko/Ja technical terms are usually transliterated from the same foreign word
- ❑ Char-level PBMT implicitly learns transliteration rules

Neural Machine Translation

❑ Ideas

(1) **Use target character sequence rather than word sequence**
- ❑ Removes the need to replace rare words with the unknown word symbol
- ❑ Simpler than other methods recently proposed to address the same issue

(2) **B-I representation of target characters**
- ❑ Better accuracy in preliminary experiment
- ❑ e.g. 結/B 果/I

(3) **Modified RNN encoder-decoder**
- ❑ Bi-directional RNN with an attention mechanism (Bahdanau et al., 2015)

$$h_t = [\overleftarrow{h}_t; \vec{h}_t]$$
$$\overleftarrow{h}_t = f_{GRU}(W_{s\_we} x_t, \overleftarrow{h}_{t+1})$$
$$\vec{h}_t = f_{GRU}(W_{s\_we} x_t, \vec{h}_{t-1})$$
$$c_t = \sum_{i=1}^{T} \alpha_{ti} h_i$$
$$\alpha_{ti} = \frac{\exp(e_{tj})}{\sum_{j=1}^{T} \exp(e_{tj})}$$
$$e_{ti} = f_{FFNN}(z_{t-1}, h_i, y_{t-1})$$

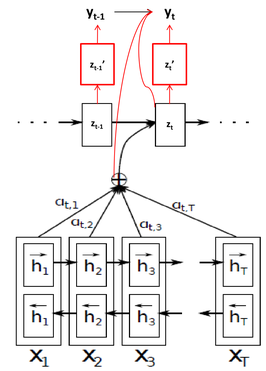
- ❑ New hidden state of the decoder

$$z_t = f_{GRU}(y_{t-1}, z_{t-1}, c_t)$$

- ❑ Prob. of next target word

$$p(y_t|y_{<t}, x) = y_t^T f_{softmax}\{W_{z'y} z'_t + W_{zy} z_t + W_{cy} c_t + W_{yy}(W_{t\_we} y_{t-1}) + b_y\}$$
$$z'_t = f_{ReLU}(W_{zz'} z_t)$$

❑ Our settings
- ❑ #Target word vocab.: 173K(En-Ja), 128K(Ko-Ja)
- ❑ #Target char. Vocab.: 6K(En-Ja), 5K(Ko-Ja)
- ❑ Src/Tgt embedding dimension: 200
- ❑ #Recurrent units: 1,000
- ❑ Optimization: Stochastic gradient descent (SGD)
- ❑ Not use dropout
- ❑ Early-stopped to maximize BLEU on dev. data



< Modified RNN Encoder-Decoder>

Experimental Results

❑ En-Ja SMT

SYS	BLEU	#Rules
Tree-to-string SB	31.34	250M
+ Rule augmentation	32.48	1950M
+ Parameter modification	32.63	1950M
+ OOV handling	32.76	1950M

- ❑ Rule augmentation increases both BLEU and #Rules
- ❑ Hyphen word split and spell correction improve the performance

❑ Ko-Ja SMT

SYS	BLEU	#Rules
Word PB	70.36	57M
Char PB	70.31	55M
Word PB + Char PB	70.91	57M & 55M

- ❑ Character-level PB is comparable to Word-level PB
- ❑ Combined system is the best

❑ Experiment with NMT variations

NMT Model	En-Ja BLEU
RNN E.-D. using target word seq.	29.78
RNN E.-D. using target char. seq.	31.25
RNN E.-D. using target char. seq. with B-I	32.05
Modified RNN E.-D. using target char. seg. with B/I	33.14

❑ Effect of combination between T2S/PBMT and NMT

SYS	En-Ja		Ko-Ja	
	BLEU	Human	BLEU	Human
T2S/PBMT only	32.76	N/A	70.91	6.75
NMT only	33.14	48.50	65.72	N/A
T2S/PBMT + NMT reranking	34.60	53.25	71.38	14.75

- ❑ NMT outperform T2S in En-Ja, while it does not outperform PBMT in Ko-Ja
- ❑ NMT reranking give a great synergy between T2S/PB and NMT