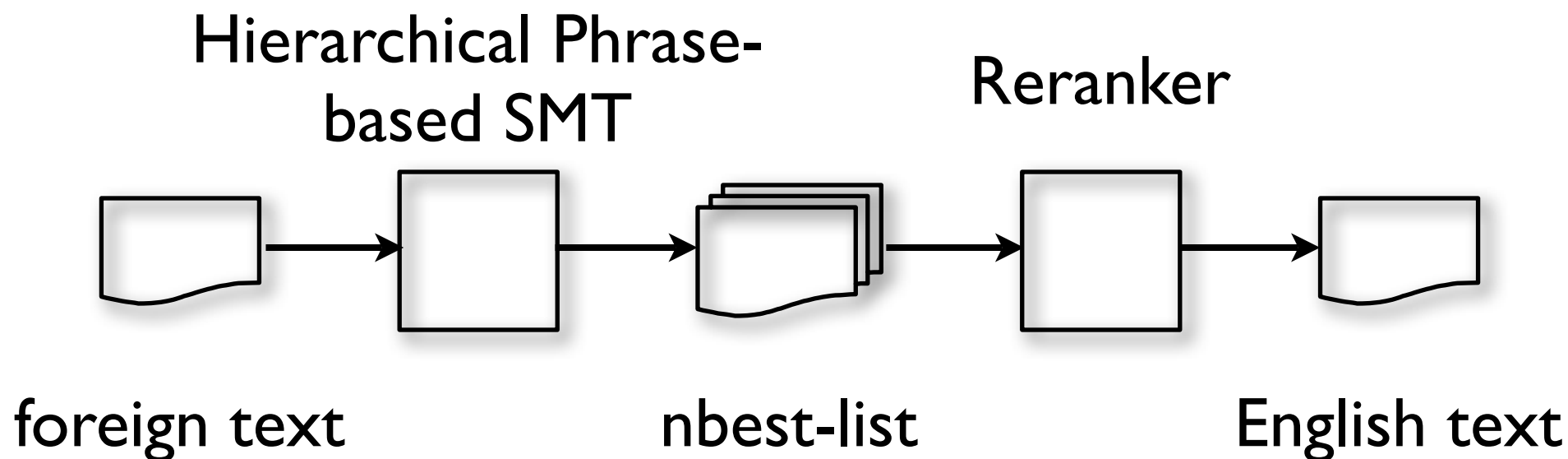


# Larger Feature Set Approach for MT: IWWSLT 2007

Taro Watanabe, Jun Suzuki, Katsuhito Sudoh,  
Hajime Tsukada and Hideki Isozaki

NTT Communication Science Labs.  
{taro,jun,sudoh,tsukada,isozaki}@cslab.kecl.ntt.co.jp

# NTT SMT System



Decoder maximizes:

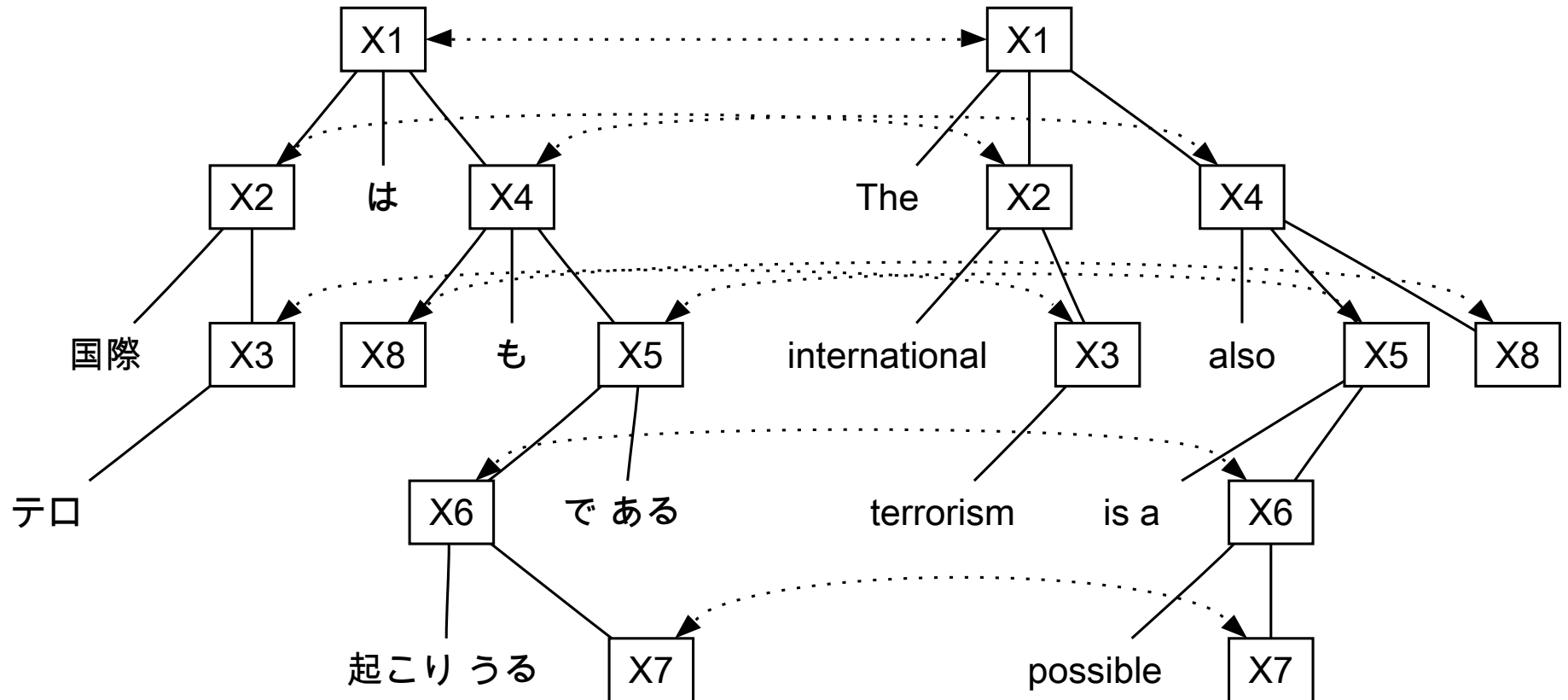
$$\hat{e} = \operatorname{argmax}_e \mathbf{w}^\top \cdot \mathbf{h}(f, e)$$

Reranker votes:

$$\hat{e} = \operatorname{argmax}_e \left\{ \mathbf{w}_i^\top \cdot \mathbf{h}(f, e) \right\}_{i=1}^n$$

Both systems employ large # of sparse features

# Hierarchical SMT



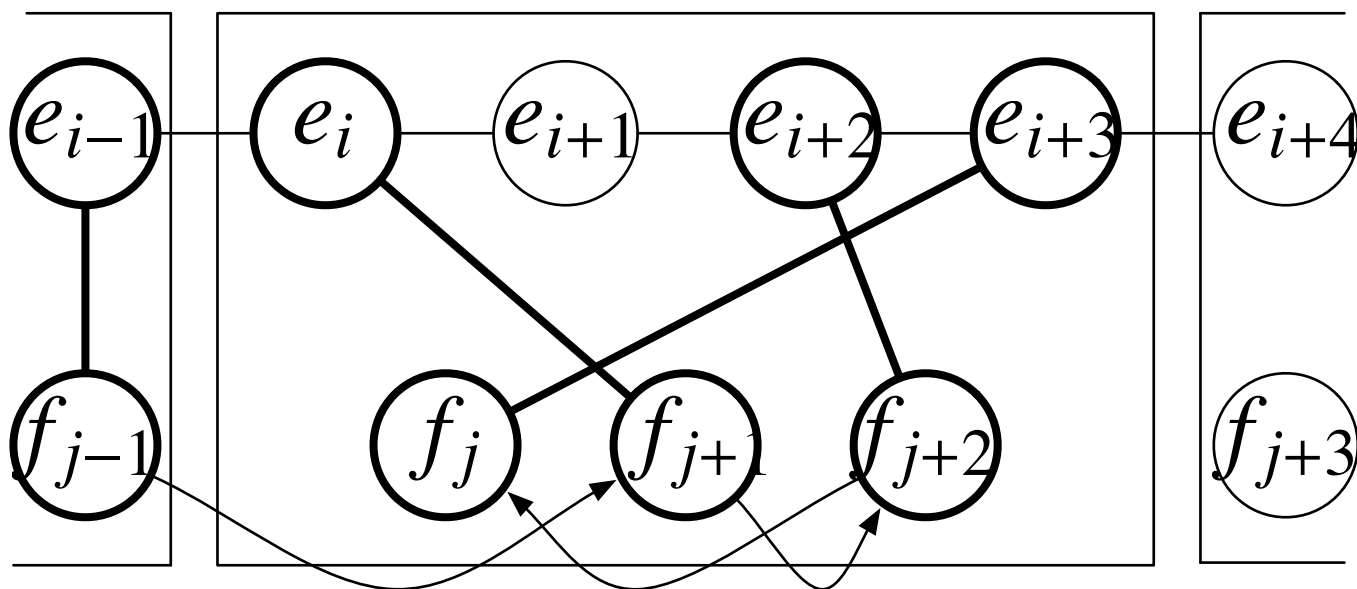
- Hierarchically embedded phrases (Chiang, 2005)
- An efficient top-down search (Watanabe et al., 2006)

# Feature Set

- 5-gram language model
- Phrase probabilities
- Lexical weights
- Insertion/deletion penalties
- # of words/phrases

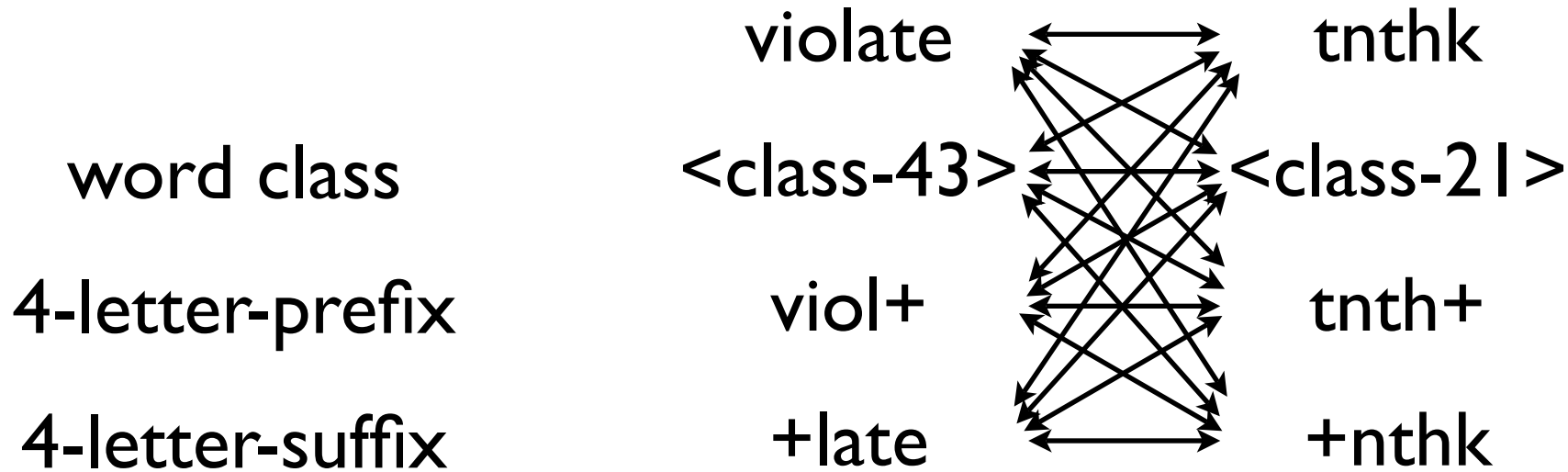
+ Sparse Features

# Sparse Features



- Preserve word alignment inside hierarchical phrases
- Word-wise features (word-pair, target-bigram etc.)

# Factoring



- Use of normalized tokens (POS/word class/prefix/etc.)
- Consider all possible combinations
  - POS: expanded into all possible solutions

# Sparse Features

- Sparse features:
  - {1,2}-gram of word-pairs
  - target word bigram
  - Insertion/deletion features
  - Hierarchical dependency features
- Word Factoring:
  - Surface word
  - Word class
  - POS/NE
  - WordNet's synset
  - 4-letter prefix/suffix

# Online Training

Training data:  $\mathcal{T} = \{(f^t, \mathbf{e}^t)\}_{t=1}^T$

$m$ -best oracles:  $\mathcal{O} = \{\}_{t=1}^T$

$i = 0$

- 1: **for**  $n = 1, \dots, N$  **do**
- 2:     **for**  $t = 1, \dots, T$  **do**
- 3:          $C^t \leftarrow \text{best}_k(f^t; \mathbf{w}^i)$
- 4:          $\mathcal{O}^t \leftarrow \text{oracle}_m(\mathcal{O}^t \cup C^t; \mathbf{e}^t)$
- 5:          $\mathbf{w}^{i+1} = \text{update } \mathbf{w}^i \text{ using } C^t \text{ w.r.t. } \mathcal{O}^t$
- 6:          $i = i + 1$
- 7:     **end for**
- 8: **end for**
- 9: **return**  $\frac{\sum_{i=1}^{NT} \mathbf{w}^i}{NT}$



# Large Margin Constraints

$$\hat{\mathbf{w}}^{i+1} = \operatorname{argmin}_{\mathbf{w}^{i+1}} \frac{1}{2} \|\mathbf{w}^{i+1} - \mathbf{w}^i\|^2 + C \sum_{\hat{e}, e'} \xi(\hat{e}, e')$$

subject to

$$s^{i+1}(f^t, \hat{e}) - s^{i+1}(f^t, e') + \xi(\hat{e}, e') \geq L(\hat{e}, e'; \mathbf{e}^t)$$

$$\xi(\hat{e}, e') \geq 0$$

$$\forall \hat{e} \in \mathcal{O}^t, \forall e' \in \mathcal{C}^t$$

- Constrained by m-oracle + k-best.
- “C” to control the amount of updates.

**Reranker**

# Reranking

## Perceptron Training

Training data:  $\mathcal{T} = \{(f^t, C^t, \mathbf{e}^t)\}_{t=1}^T$

```
1: for  $n = 1, \dots, N$  do
2:    $\mathbf{w}^n = \mathbf{w}^{n-1}$ 
3:   for  $t = 1, \dots, T$  do
4:      $\mathcal{R} = \text{rerank}(C^t; \mathbf{w}^n)$ 
5:     for  $i = 1, \dots, |\mathcal{R}|$  do
6:       for  $j = i + 1, \dots, |\mathcal{R}|$  do
7:         if  $L(\mathcal{R}_j, \mathcal{R}_i; \mathbf{e}^t) > 0$  then
8:            $\mathbf{w}^n = \text{update } \mathbf{w}^n \text{ using } \mathcal{R}_i \text{ and } \mathcal{R}_j$ 
9:         end if
10:      end for
11:    end for
12:  end for
13: end for
14: return  $\{\mathbf{w}^n\}_{n=1}^N$ 
```

## Decoding (Voting)

$k$ -best translation list:  $(f, C)$

Weight vectors:  $\{\mathbf{w}^n\}_{n=1}^N$

Votes:  $\mathcal{V} = \mathbf{0}$

```
1: for  $n = 1, \dots, N$  do
2:    $\hat{i} = \text{argmax}_i \{\mathbf{w}^n\}^\top \cdot \mathbf{h}(f, C_i)$ 
3:    $\mathcal{V}_{\hat{i}} = \mathcal{V}_{\hat{i}} + 1$ 
4: end for
5: return  $C_{\hat{i}}$  where  $\hat{i} = \text{argmax}_i \mathcal{V}_i$ 
```

## Parameter Update

$$\mathbf{w}^n = \mathbf{w}^n + L(\mathcal{R}_j, \mathcal{R}_i; \mathbf{e}^t) \cdot (\mathbf{h}(f^t, \mathcal{R}_j) - \mathbf{h}(f^t, \mathcal{R}_i))$$

# Objectives

- Document-BLEU or sentence-BLEU?

$$\text{BLEU}(E; \mathbf{E}) = \exp\left(\frac{1}{N} \sum_{n=1}^N \log p_n(E, \mathbf{E})\right) \cdot \text{BP}(E, \mathbf{E})$$

- Our method: compute the difference from an oracle BLEU (Watanabe et al., 2006)

$$\text{BLEU}(\{\hat{e}^1, \dots, \hat{e}^{t-1}, e', \hat{e}^{t+1}, \dots, \hat{e}^T\}; \mathbf{E})$$

- Loss by an approximated BLEU  $\approx$  document-wise loss.

# Task Setting

# Preprocessing

- Removed bitexts matching regexp: [0-9]
- English: MaxEnt/Brill POS tagger
- Arabic: Isolate Arabic scripts/punctuations
- Italian: Treetagger
- Japanese/Chinese: HMM-based POS/NE tagger
- Casing preserved for English
- Punctuation removed for source side

# Bitexts

	ar-en	it-en	ja-en	zh-en
sentences	833K	854K	1.0M	3.3M
words	25M	24M	8.6M	57M
vocabulary	132K	67K	254K	961K
source	LDC	EuroParl	NiCT	LDC

- Data comes from various sources (LDC or public domain)
- We used devset 4,5,5b for tuning, since they had ASR data.

# Task Adaptation

Source side 3-gram perplexity

	ar-en	it-en	ja-en	zh-en
dev 4,5,5b	561.96	277.24	51.29	188.49
test	214.99	271.39	13.45	73.18

- Sample bitexts for phrase-table extraction (Ittycheriah and Roukos, 2007)
- For each source sentence in test(dev) set:
  - Extract bitexts from the universe of training data.
  - Similarity measured by ngram precision.



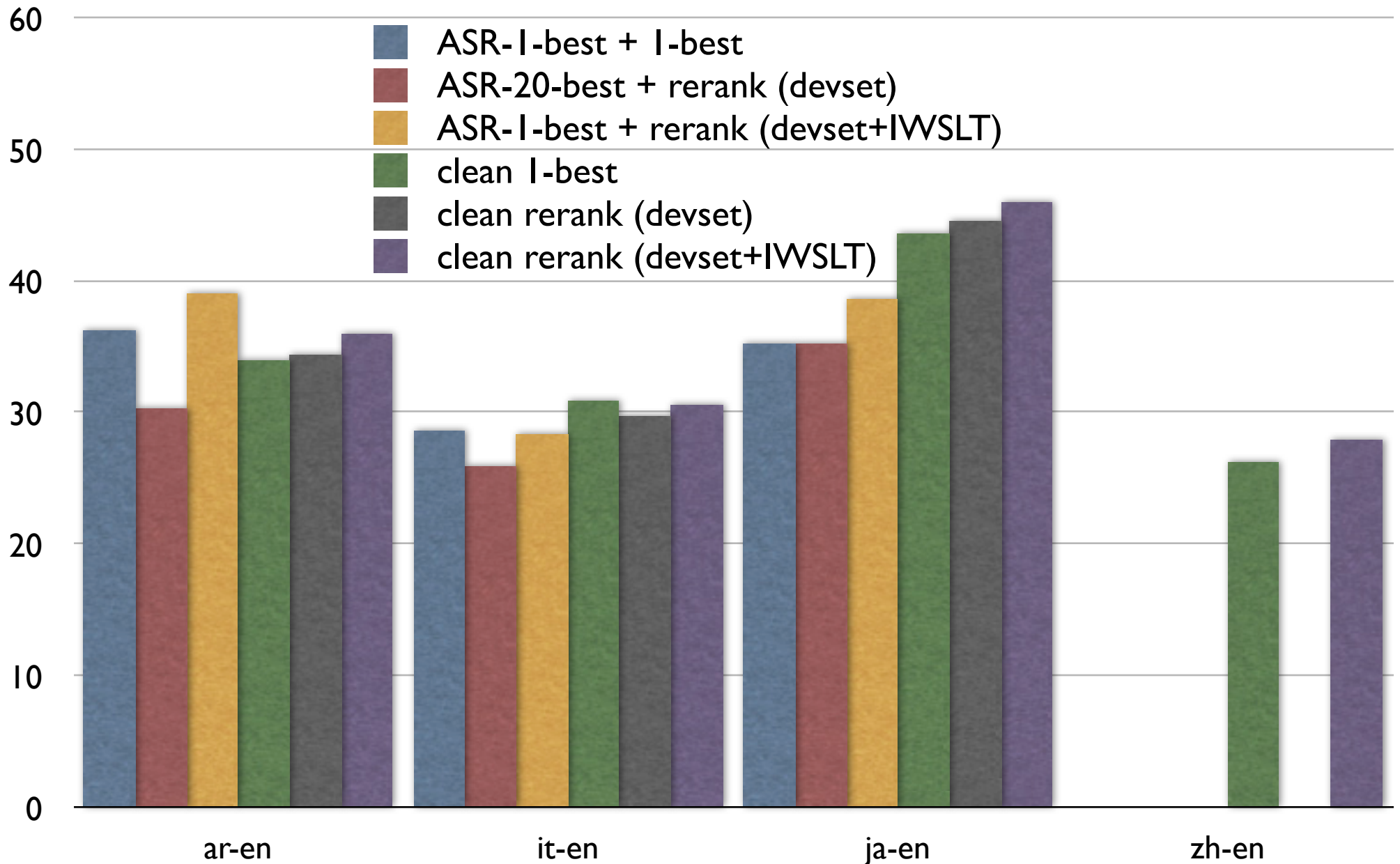
# ASR Translation

- 1-best ASR translation
- 20-best ASR translation
  - Translate all the 20-bests and select the best one by our reranker.
  - Various word/sentence-wise confidence measures integrated as features.

# Parameter Estimation

- Decoder:
  - Estimated on devset 4, 5, 5b.
  - 200-300 iterations
- Reranker:
  - 1,000-best list
  - Estimated on devset 4, 5, 5b and IWSLT's 20,000 sentences.

# Results (BLEU)

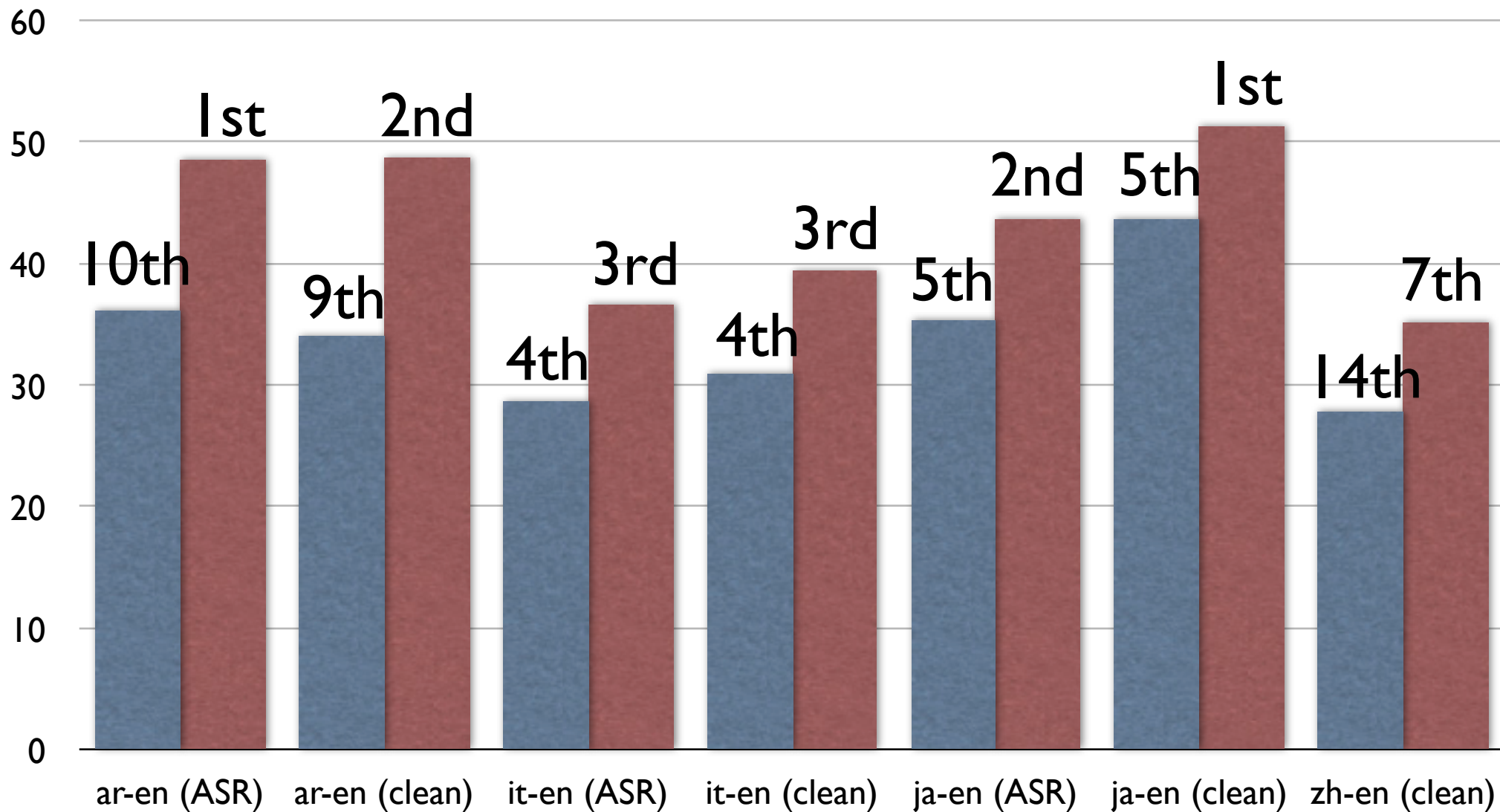


# Post Evaluation

- Use IWSLT data only.....
- Held-out set to terminate iterations
- Arabic/Japanese/Chinese are close to IWSLT data.
  - Estimated on devset 1 and 2, held-out devset 3.
- Italian data is totally different:
  - Extract phrases from devset 5b, too
  - Estimation on devset 4 and 5, held-out devset 5b

# Results (BLEU)

Primary Post-Evaluation



# Conclusion

- NTT SMT System:
  - Large # of features are integrated both in decoder/reranker
  - Careful devset selection
  - Careful tuning
  - Larger data helps for reranking
- Future Work:
  - More rich features, more experiments.