北京交通大學
BEIJING JIAOTONG UNIVERSITY

Improved Statistical Machine Translation with Source Language Paraphrase

Chen Su, Yujie Zhang, Jin’an Xu, Zhen Guo

Beijing Jiaotong University

北京交通大學 BEIJING JIAOTONG UNIVERSITY

Outline

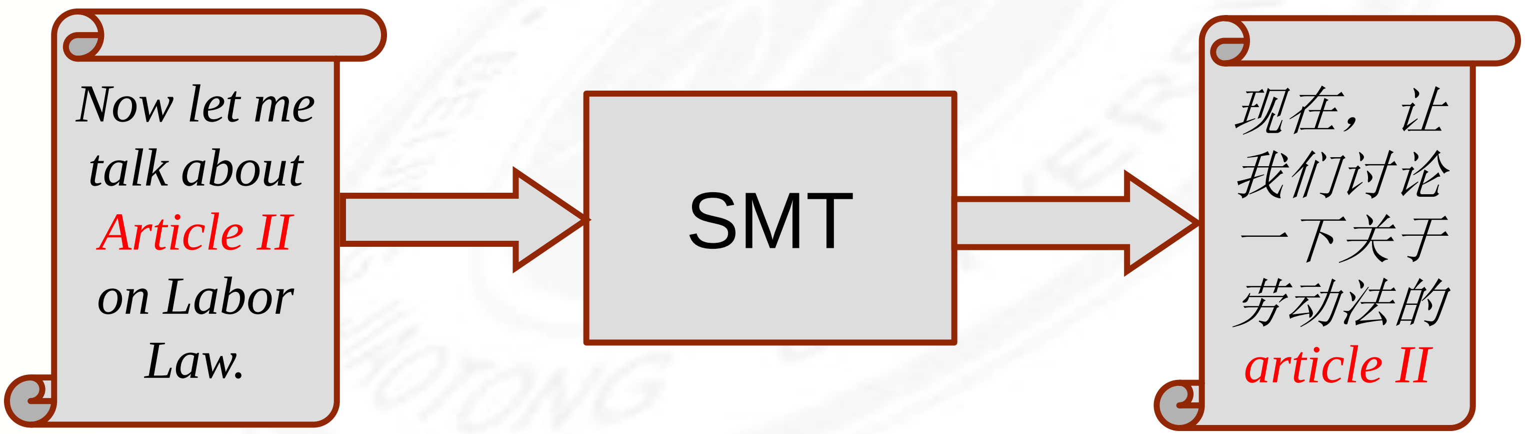
- Background
- Investigation & Research
- Decoding
- Experiment & Result

Background

- SMT system needs training data
  - Parallel corpus
  - Large-scale, Wide-cover
- Existing problems
  - It is difficult to obtain large scale parallel corpus

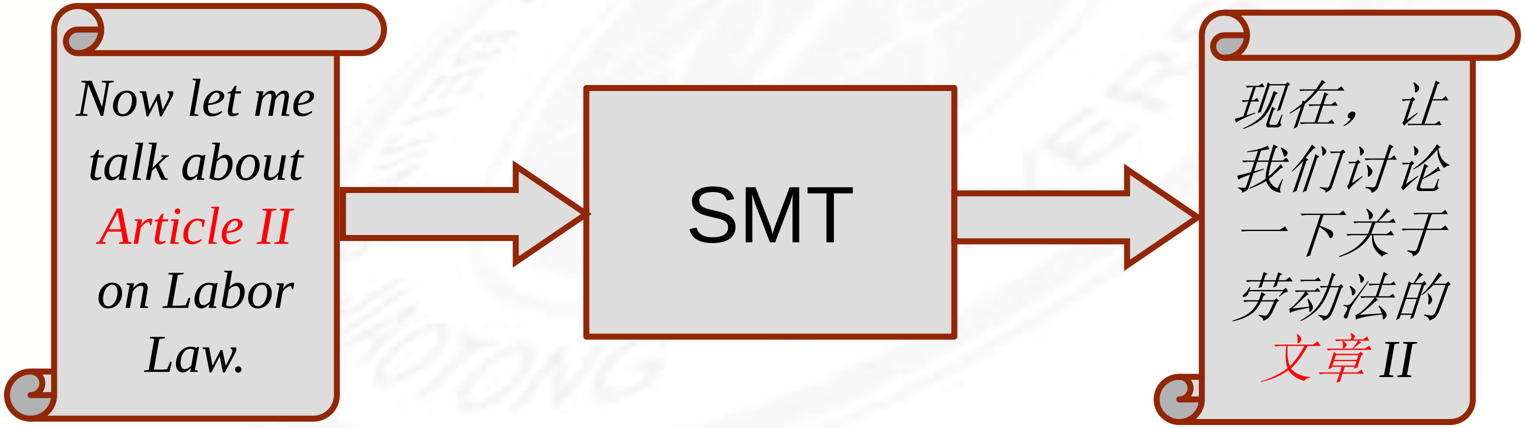
Common problems(1/2)

- Out-Of-Vocabulary (OOV)
  - SMT system didn't do anything for OOVs, so they are retained in translation, which affect translation quality.

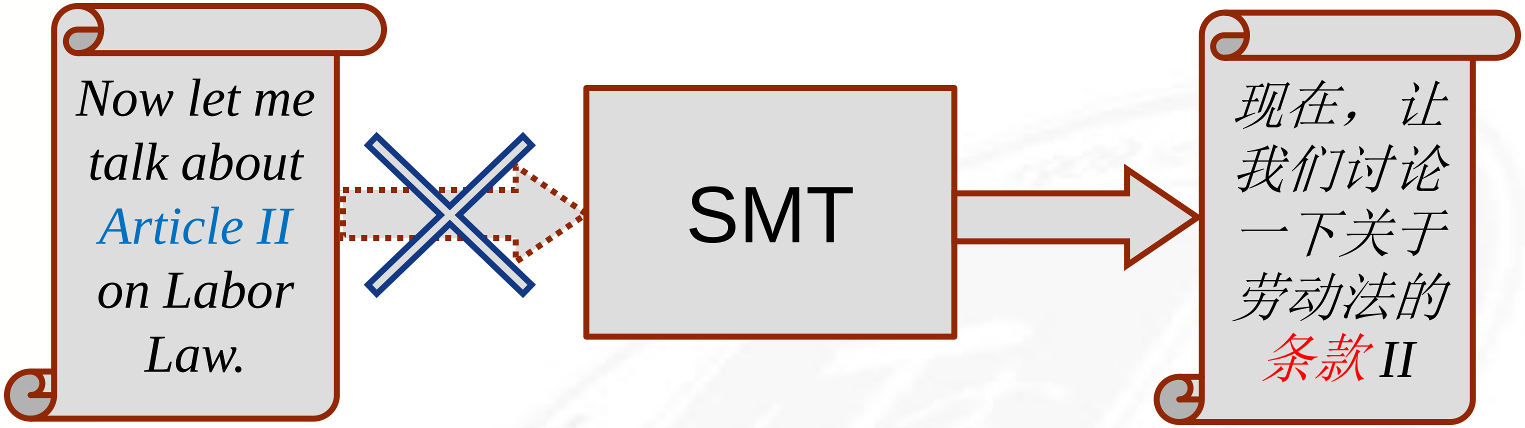


Common problems(2/2)

- Lacking abundant candidate translations
  - Generally speaking, phrase translation table can't cover all translation knowledge for every phrase, which causes the test sentences cannot be translated properly.

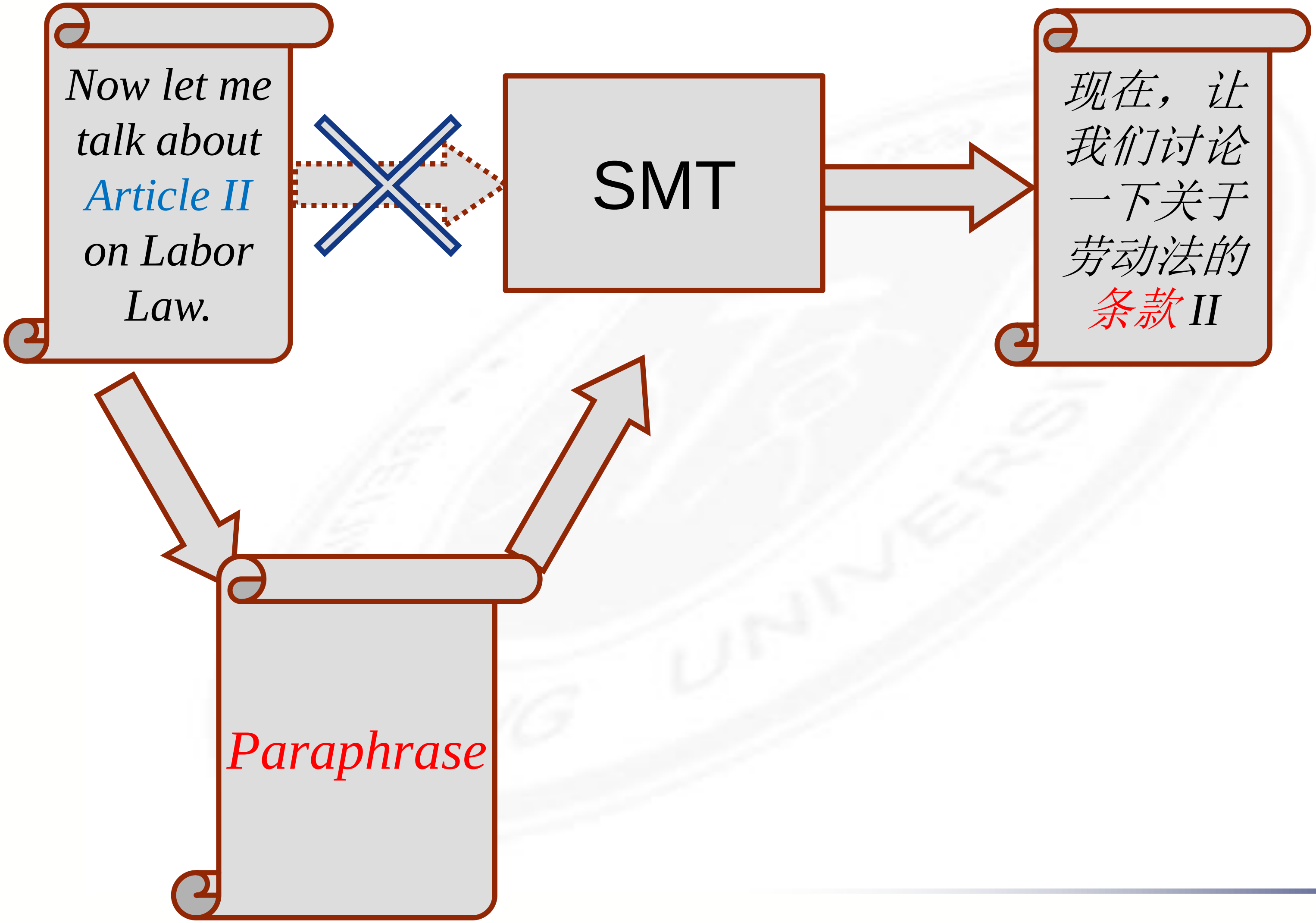


Solution(1/2)



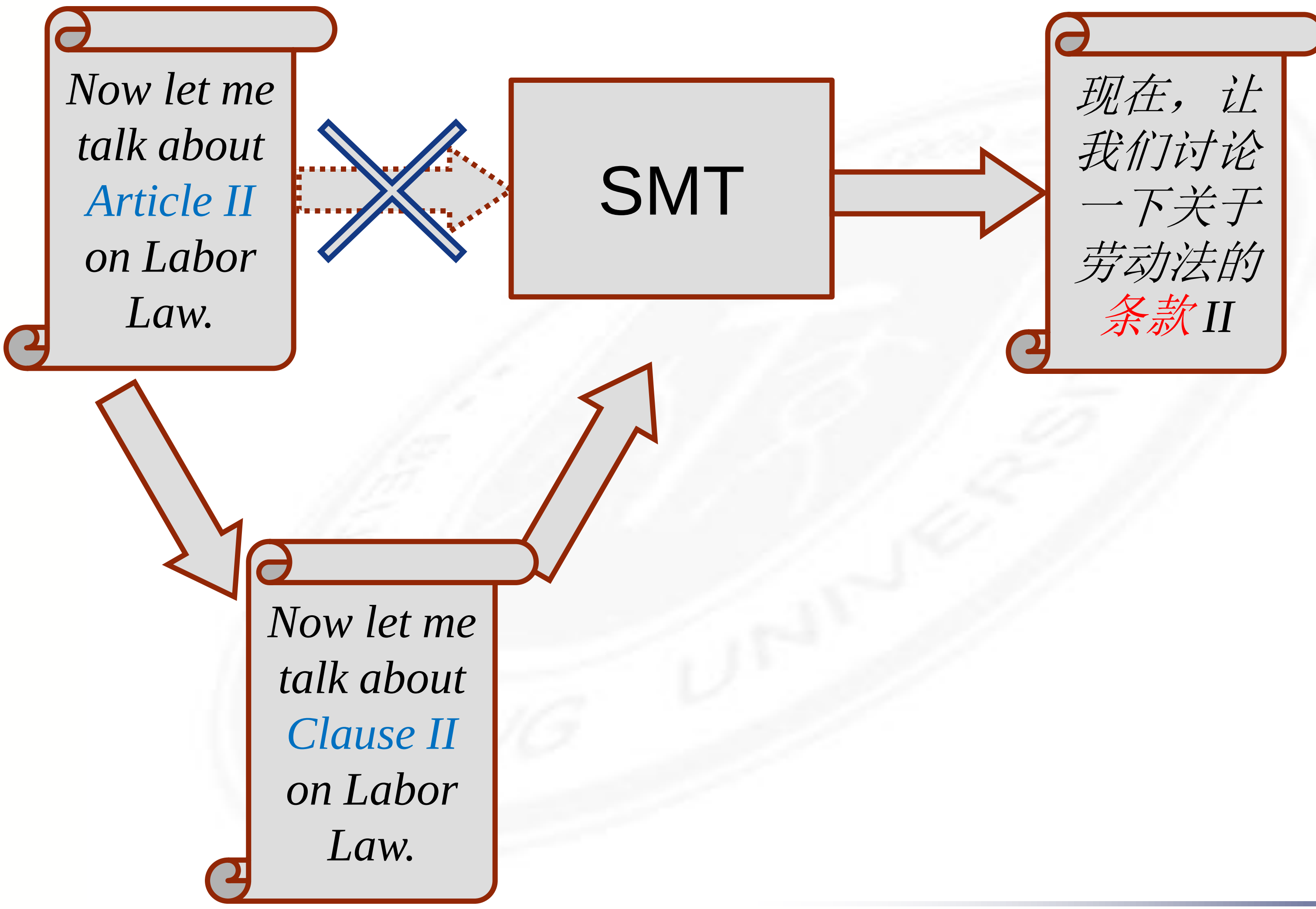
Now let me talk about Article II on Labor Law.
SMT
现在，让我们讨论一下关于劳动法的条款II

Solution(1/2)



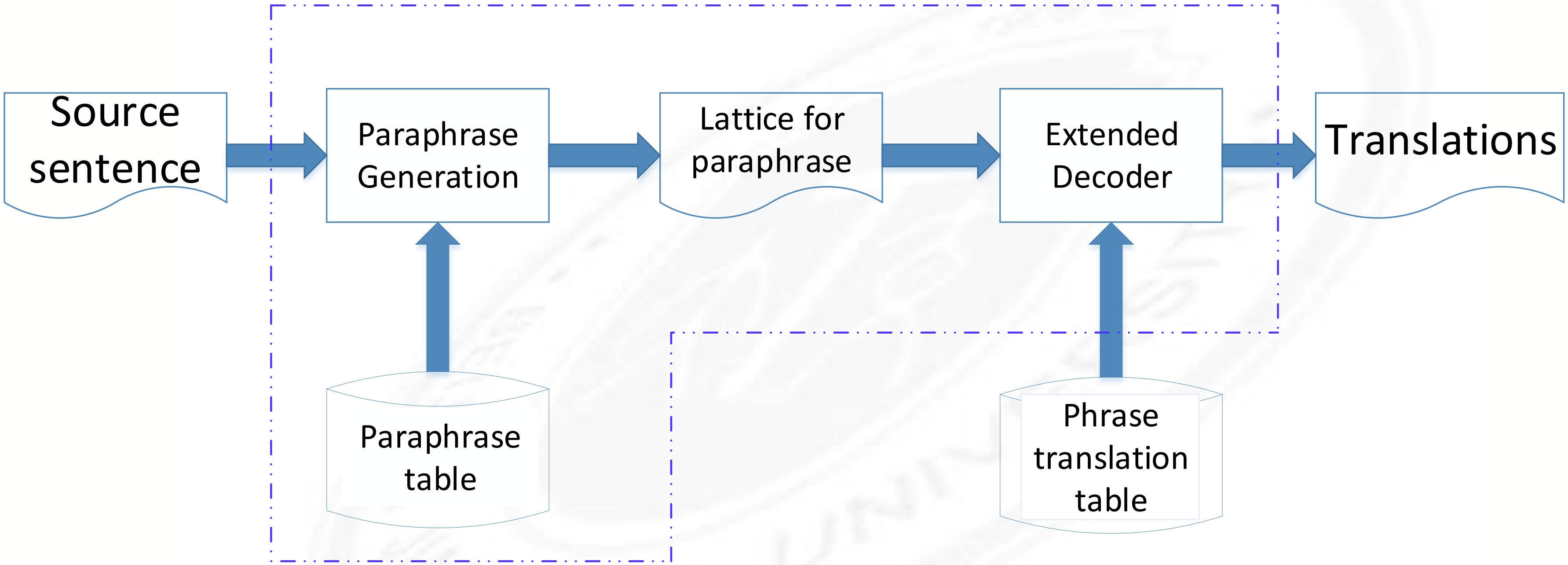
Now let me talk about Article II on Labor Law.
SMT
现在，让我们讨论一下关于劳动法的条款II
Paraphrase

Solution(1/2)


Now let me talk about Article II on Labor Law.
SMT
现在，让我们讨论一下关于劳动法的条款II
Now let me talk about Clause II on Labor Law.

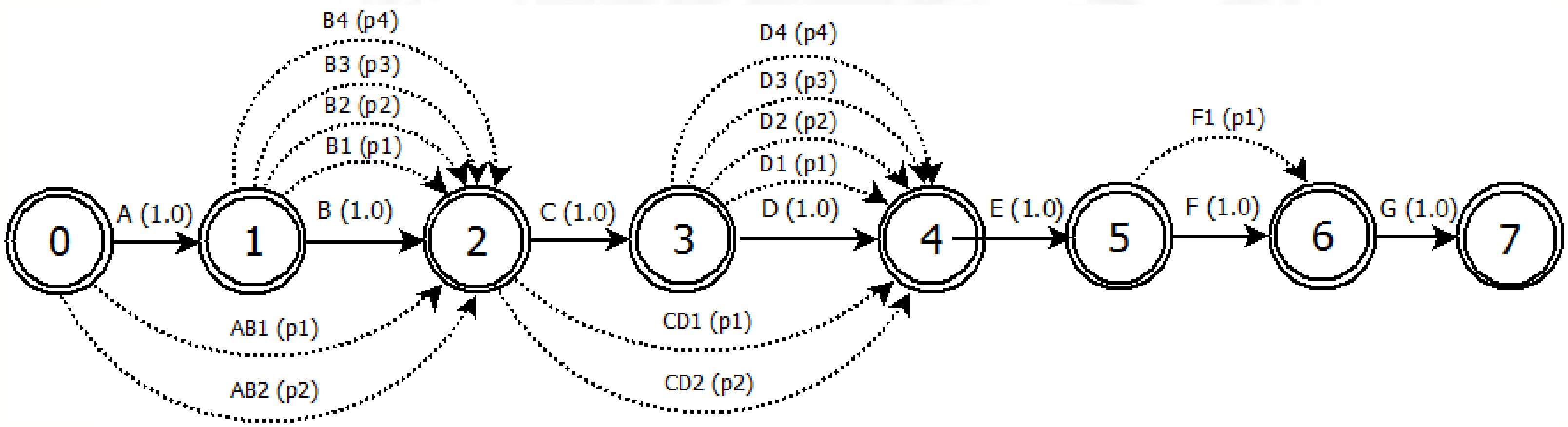
Solution(2/2)

- Framework: SMT with paraphrase



Related work

- Integrating paraphrase knowledge for SMT(Du Jinhua 2010)
  - Use lattice graph to denote input sentence‘s different paraphrases.
  - Du’s work adopt heuristic method for estimate the weight of paraphrase. (**Fixed weights**)



Acquirement of paraphrase

- Obtain paraphrases by pivoting with additional bilingual corpora

English → Japanese

……

Push-bike→自転車
bike→自転車
bicycle→自転車

……

$$para(e_2|e_1) \approx \sum_{jp} p(e_2|jp) \cdot p(jp|e_1)$$

$$p(e_2|jp) \approx \frac{count(e_2, jp)}{\sum_{e_2} count(e_2, jp)}$$

$$p(jp|e_1) \approx \frac{count(jp, e_1)}{\sum_{jp} count(jp, e_1)}$$

Investigation(1/5)

- Purpose
  - How the coverage of OOV is improved with paraphrase
  - How the translation is improved with paraphrase
- Experimental data
  - NTCIR-10 English-Chinese corpus
    - Used for acquiring translation knowledge
  - NTCIR-10 English-Japanese corpus
    - Used for acquiring paraphrase knowledge

Investigation(2/5)

Experiment & Evaluation for coverage of OOV

- Divide the training data into 3 different part with different scale, and then acquire translation knowledge.
- Compare coverage of N-gram before and after adding paraphrase.

Number of N-gram on Testing data	Coverage (Phrase translation→ Phrase translation with paraphrase)		
	10K Training Data	100K Training Data	1M Training Data
1元(6,274)	77.19%→89.97%	91.07%→93.27%	95.41%→95.55%
2元(28,993)	35.57%→67.52%	62.95%→74.08%	80.15%→82.59%
3元(42,937)	12.97%→37.11%	30.01%→42.63%	48.91%→53.09%
4元(46,974)	4.20%→14.94%	11.50%→18.39%	22.77%→25.91%
5元(47,316)	1.50%→5.55%	4.62%→7.36%	10.66%→11.98%
6元(46,389)	0.55%→2.12%	1.99%→3.09%	5.36%→5.99%
7元(44,918)	0.24%→0.81%	0.97%→1.37%	3.08%→3.31%

Investigation(3/5)

- Experiment: Coverage of correct translation
  - Definition:
    - ***Similarity*** : longest common subsequence between translation and reference, which is normalized by length of reference translation.(the unit is character)
    - ***Ideal translation*** : the candidate translation with the highest similarity score.
  - Objects: evaluate changes of the similarity between ideal translation and reference without/with paraphrase.
  - Methods: search the ideal translation with CKY algorithm.

北京交通大學 BEIJING JIAOTONG UNIVERSITY

Investigation(4/5)

Longest common subsequence between English's phrase e_i^j's ideal translation and reference c_l^m

$$f(e_i^j, c_l^m) = \max \begin{cases} Length(c_l^m) & \text{if } e_i^j \to c_l^m \text{ is existing in phrase table(PT)} \\ f(e_i^k, c_l^n) + f(e_k^j, c_n^m) & (i < k < j, l < n < m) \text{ monotone order} \\ f(e_i^k, c_n^m) + f(e_k^j, c_l^n) & (i < k < j, l < n < m) \text{ swap order} \\ f(e_{i+1}^j, c_l^m) & \text{if } e_i^{i+1} \to null \text{ is existing in PT} \\ f(e_i^{j-1}, c_l^m) & \text{if } e_{j-1}^j \to null \text{ is existing in PT} \\ f(e_i^j, c_{l+1}^m) & \text{if } c_l^{l+1} \to null \text{ is existing in PT} \\ f(e_i^j, c_l^{m-1}) & \text{if } c_{m-1}^m \to null \text{ is existing in PT} \end{cases}$$

Similarity describes the similarity of whole testing data, and S is the number of sentences of the testing data.

$$Similarity(c, c_{ref}) = \sum_{s=1}^{S} f(e_s, c_s) \Big/ \sum_{s=1}^{S} Length(c_s)$$

北京交通大學
BEIJING JIAOTONG UNIVERSITY

Investigation(5/5)

- Changes of the similarity without/with paraphrase

Translation resource	Similarity		
	10K Training data	100K Training data	1M Training data
Phrase translation table	82.82%	91.22%	94.31%
Phrase translation table + paraphrase knowledge	92.46%	95.88%	97.03%

SMT decoding algorithm employing paraphrase(1/2)

- Decoding for maximum-entropy SMT model

$$\hat{c} = \arg\max_{c}\{\Pr(c|e_1)\} = \arg\max_{c}\{\sum_{m=1}^{M}\lambda_m h_m(c,e_1)\}$$

- Features about phrase via paraphrase
  - (In order/Reverse) phrase translation features

$$\hat{h}_{Tran}(c,e_1) = \log \hat{p}(c|e_1) = \log\left[para(e_2|e_1)^{\alpha_1} \cdot p(c|e_2)\right]$$

$$\hat{h}_{VerTran}(c,e_1) = \log \hat{p}(e_1|c) = \log\left[para(e_1|e_2)^{\alpha_2} \cdot p(e_2|c)\right]$$

  - (In order/Reverse) lexical translation features

$$\hat{h}_{Lex}(c,e_1) = \log \hat{L}ex(c|e_1) = \log\left[para(e_2|e_1)^{\alpha_3} \cdot Lex(c|e_2)\right]$$

$$\hat{h}_{VerLex}(c,e_1) = \log \hat{L}ex(e_1|c) = \log\left[para(e_1|e_2)^{\alpha_4} \cdot Lex(e_2|c)\right]$$

北京交通大學 BEIJING JIAOTONG UNIVERSITY

SMT decoding algorithm employing paraphrase(2/2)

- Formula transformation:
  - two new paraphrase features are added.

$$
\begin{aligned}
&\lambda_{Tran}\cdot\hat{h}_{Tran}(c,e_1)+\lambda_{Lex}\cdot\hat{h}_{Lex}(c,e_1)\\
&=\lambda_{Tran}\log p(c\,|\,e_2)+\lambda_{Lex}\log Lex(c\,|\,e_2)+(\lambda_{Tran}\cdot\alpha_1+\lambda_{Lex}\cdot\alpha_3)\log para(e_2\,|\,e_1)\\
&=\lambda_{Tran}\cdot h_{Tran}(c,e_2)+\lambda_{Lex}\cdot h_{Lex}(c,e_2)+\boxed{\lambda_{Para}\cdot h_{Para}(e_1,e_2)}
\end{aligned}
$$

$$
\begin{aligned}
&\lambda_{VerTran}\cdot\hat{h}_{VerTran}(c,e_1)+\lambda_{VerLex}\cdot\hat{h}_{VerLex}(c,e_1)\\
&=\lambda_{VerTran}\cdot h_{VerTran}(c,e_2)+\lambda_{VerLex}\cdot h_{VerLex}(c,e_2)+\boxed{\lambda_{VerPara}\cdot h_{VerPara}(e_1,e_2)}
\end{aligned}
$$

- Use MERT for optimizing model weight.

Experiment & Evaluation

Data

- NTCIR-10 English-Chinese corpus used for training SMT system
- NTCIR-10 English-Japanese corpus used for acquiring paraphrase

SMT system for comparison

- Baseline： traditional phrase-based SMT
- Du System： traditional phrase-based SMT with fixed weight of paraphrase lattice
- Our System： traditional phrase-based SMT with paraphrase features

Evaluation result

SMT system	BLEU score(%)		
	10K training data	100K training data	1M training data
Baseline	35.69	40.39	44.16
Du System	36.72(+1.03)	40.68(+0.29)	43.43(-0.73)
Our System	37.09(+1.40)	40.42(+0.30)	44.48(+0.32)

Conclusion

- Integrated paraphrase into SMT
  - We redesigned the paraphrase as features in SMT decoding algorithm.
- Investigated the effect of paraphrase in
  - Coverage of OOV
  - Coverage of correct translation
- Evaluated the SMT system performance
  - The improvement was proved in BLEU.

Thank you!