Semi-supervised Learning for Neural Machine Translation

Yong Jiang

Tencent AI Lab && ShanghaiTech University

Paper 1

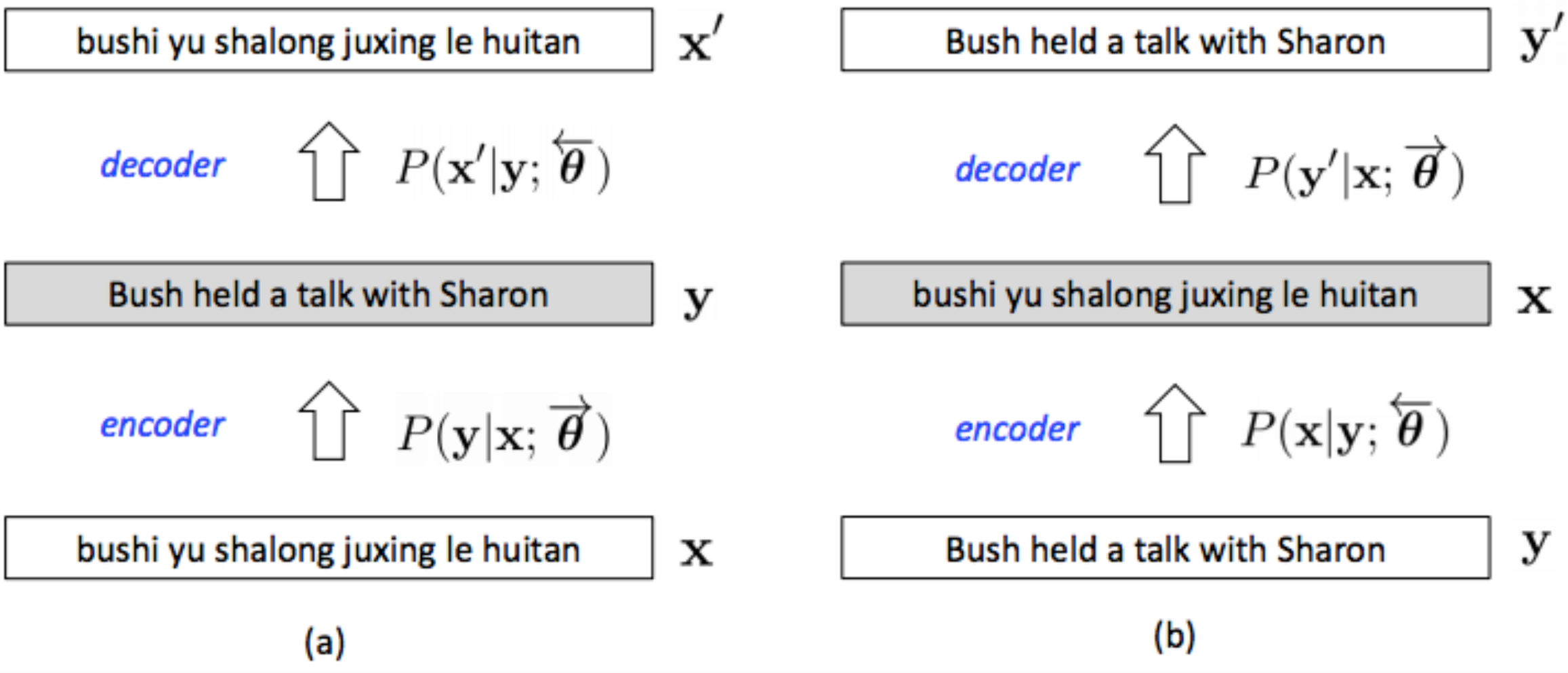
Semi-Supervised Learning for Neural Machine Translation

Yong Cheng[#], Wei Xu[#], Zhongjun He[+], Wei He[+], Hua Wu[+], Maosong Sun[†] and Yang Liu[†] *
[#]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
[†]State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing, China
[+]Baidu Inc., Beijing, China
chengyong3001@gmail.com weixu@tsinghua.edu.cn
{hezhongjun,hewei06,wu_hua}@baidu.com
{sms,liuyang2011}@tsinghua.edu.cn

Motivation

- **Free** monolingual available data is everywhere

Model


bushi yu shalong juxing le huitan
$\mathbf{x}'$
decoder
$P(\mathbf{x}'|\mathbf{y};\overleftarrow{\boldsymbol{\theta}})$
Bush held a talk with Sharon
$\mathbf{y}$
Bush held a talk with Sharon
$\mathbf{y}'$
decoder
$P(\mathbf{y}'|\mathbf{x};\overrightarrow{\boldsymbol{\theta}})$
bushi yu shalong juxing le huitan
$\mathbf{x}$
encoder
$P(\mathbf{y}|\mathbf{x};\overrightarrow{\boldsymbol{\theta}})$
bushi yu shalong juxing le huitan
$\mathbf{x}$
encoder
$P(\mathbf{x}|\mathbf{y};\overleftarrow{\boldsymbol{\theta}})$
Bush held a talk with Sharon
$\mathbf{y}$

(a) (b)

Model

- Parameters
  - params of seq2seq from x to y
  - params of seq2seq from y to x
- Objective function

$$
\begin{aligned}
& P(\mathbf{y}'|\mathbf{y}; \overrightarrow{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}}) \\
= & \sum_{\mathbf{x}} P(\mathbf{y}', \mathbf{x}|\mathbf{y}; \overrightarrow{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}}) \\
= & \sum_{\mathbf{x}} \underbrace{P(\mathbf{x}|\mathbf{y}; \overleftarrow{\boldsymbol{\theta}})}_{\textit{encoder}} \underbrace{P(\mathbf{y}'|\mathbf{x}; \overrightarrow{\boldsymbol{\theta}})}_{\textit{decoder}}
\end{aligned}
$$

$$
\begin{aligned}
& P(\mathbf{x}'|\mathbf{x}; \overrightarrow{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}}) \\
= & \sum_{\mathbf{y}} P(\mathbf{x}', \mathbf{y}|\mathbf{x}; \overleftarrow{\boldsymbol{\theta}}) \\
= & \sum_{\mathbf{y}} \underbrace{P(\mathbf{y}|\mathbf{x}; \overrightarrow{\boldsymbol{\theta}})}_{\textit{encoder}} \underbrace{P(\mathbf{x}'|\mathbf{y}; \overleftarrow{\boldsymbol{\theta}})}_{\textit{decoder}}
\end{aligned}
$$

$$
\begin{aligned}
& J(\overrightarrow{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}}) \\
= & \underbrace{\sum_{n=1}^{N} \log P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}; \overrightarrow{\boldsymbol{\theta}})}_{\textit{source-to-target likelihood}} \\
& + \underbrace{\sum_{n=1}^{N} \log P(\mathbf{x}^{(n)}|\mathbf{y}^{(n)}; \overleftarrow{\boldsymbol{\theta}})}_{\textit{target-to-source likelihood}} \\
& + \lambda_1 \underbrace{\sum_{t=1}^{T} \log P(\mathbf{y}'|\mathbf{y}^{(t)}; \overrightarrow{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}})}_{\textit{target autoencoder}} \\
& + \lambda_2 \underbrace{\sum_{s=1}^{S} \log P(\mathbf{x}'|\mathbf{x}^{(s)}; \overrightarrow{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}})}_{\textit{source autoencoder}}
\end{aligned}
$$

Optimization

$$\overrightarrow{\boldsymbol{\theta}}^* = \text{argmax} \left\{ \sum_{n=1}^{N} \log P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}; \overrightarrow{\boldsymbol{\theta}}) + \right.$$
$$\lambda_1 \sum_{t=1}^{T} \log P(\mathbf{y}'|\mathbf{y}^{(t)}; \overrightarrow{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}}) +$$
$$\left. \lambda_2 \sum_{s=1}^{S} \log P(\mathbf{x}'|\mathbf{x}^{(s)}; \overrightarrow{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}}) \right\}$$

$$\overleftarrow{\boldsymbol{\theta}}^* = \text{argmax} \left\{ \sum_{n=1}^{N} \log P(\mathbf{x}^{(n)}|\mathbf{y}^{(n)}; \overleftarrow{\boldsymbol{\theta}}) + \right.$$
$$\lambda_1 \sum_{t=1}^{T} \log P(\mathbf{y}'|\mathbf{y}^{(t)}; \overrightarrow{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}}) +$$
$$\left. \lambda_2 \sum_{s=1}^{S} \log P(\mathbf{x}'|\mathbf{x}^{(s)}; \overrightarrow{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}}) \right\}$$

$$\frac{\partial J(\overrightarrow{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}})}{\partial \overrightarrow{\boldsymbol{\theta}}}$$
$$= \sum_{n=1}^{N} \frac{\partial \log P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}; \overrightarrow{\boldsymbol{\theta}})}{\partial \overrightarrow{\boldsymbol{\theta}}}$$
$$+\lambda_1 \sum_{t=1}^{T} \frac{\partial \log P(\mathbf{y}'|\mathbf{y}^{(t)}; \overrightarrow{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}})}{\partial \overrightarrow{\boldsymbol{\theta}}}$$
$$+\lambda_2 \sum_{s=1}^{S} \frac{\partial \log P(\mathbf{x}'|\mathbf{x}^{(s)}; \overrightarrow{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}})}{\partial \overrightarrow{\boldsymbol{\theta}}}$$

Hard!

Optimization

$$\frac{\partial \log P(\mathbf{x}'|\mathbf{x}^{(s)}; \overrightarrow{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}})}{\partial \overrightarrow{\boldsymbol{\theta}}}$$

$$\frac{\sum_{\mathbf{x}\in\mathcal{X}(\mathbf{y})} P(\mathbf{x}|\mathbf{y}; \overleftarrow{\boldsymbol{\theta}}) P(\mathbf{y}'|\mathbf{x}; \overrightarrow{\boldsymbol{\theta}}) \frac{\partial \log P(\mathbf{y}'|\mathbf{x}; \overrightarrow{\boldsymbol{\theta}})}{\partial \overrightarrow{\boldsymbol{\theta}}}}{\sum_{\mathbf{x}\in\mathcal{X}(\mathbf{y})} P(\mathbf{x}|\mathbf{y}; \overleftarrow{\boldsymbol{\theta}}) P(\mathbf{y}'|\mathbf{x}; \overrightarrow{\boldsymbol{\theta}})}$$

$$\frac{\sum_{\mathbf{x}\in\tilde{\mathcal{X}}(\mathbf{y})} P(\mathbf{x}|\mathbf{y}; \overleftarrow{\boldsymbol{\theta}}) P(\mathbf{y}'|\mathbf{x}; \overrightarrow{\boldsymbol{\theta}}) \frac{\partial \log P(\mathbf{y}'|\mathbf{x}; \overrightarrow{\boldsymbol{\theta}})}{\partial \overrightarrow{\boldsymbol{\theta}}}}{\sum_{\mathbf{x}\in\tilde{\mathcal{X}}(\mathbf{y})} P(\mathbf{x}|\mathbf{y}; \overleftarrow{\boldsymbol{\theta}}) P(\mathbf{y}'|\mathbf{x}; \overrightarrow{\boldsymbol{\theta}})}$$

select top ’10’ x in P(x|y)

Experiments: Setup

- Chinese English dataset
- Labeled dataset: LDC consists of 2.56M sentence pairs with 67.53M Chinese words and 74.81M English words. The vocabulary sizes of Chinese and English are 0.21M and 0.16M
- Unlabeled dataset: Chinese (18.75M) and English (22.32M) parts of the Xinhua portion of the GIGAWORD corpus as the monolingual corpora.
- Valid dataset: NIST 2006
- Test dataset: NIST 2002, 2003, 2004, 2005

Experiments: Beam Size

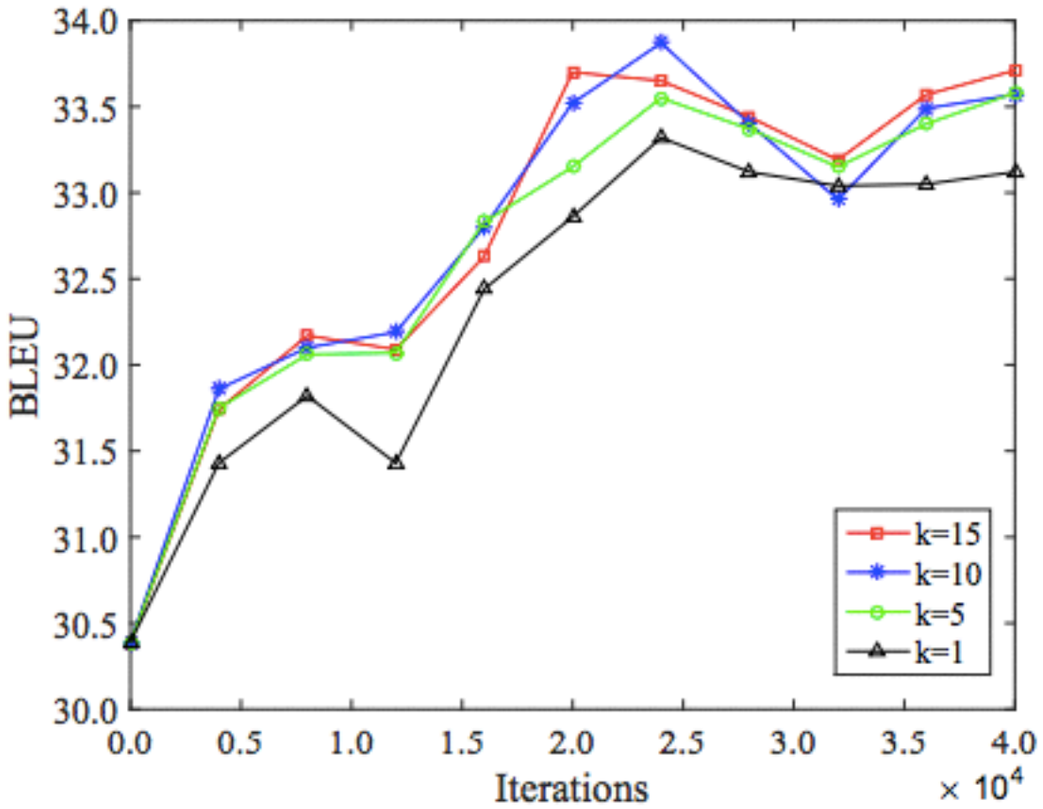


Figure 2: Effect of sample size k on the Chinese-to-English validation set.

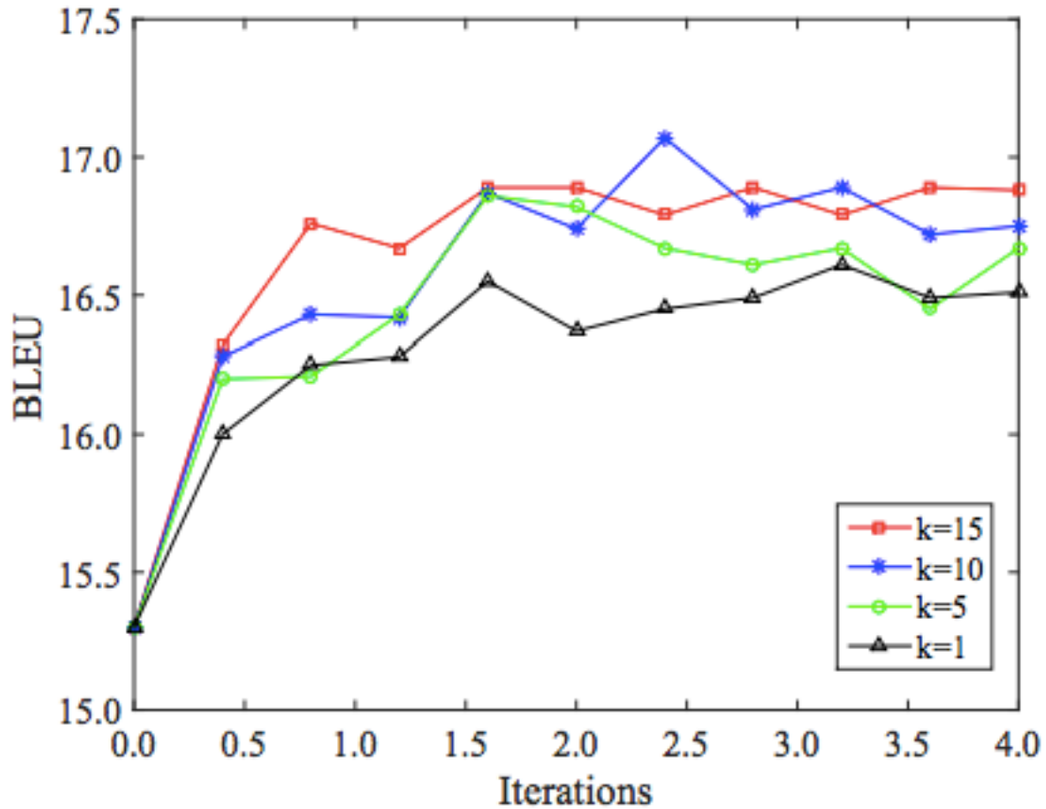


Figure 3: Effect of sample size k on the English-to-Chinese validation set.

Experiments: Results

System	Training Data			Direction	NIST06	NIST02	NIST03	NIST04	NIST05
	CE	C	E						
MOSES	√	×	×	C → E	32.48	32.69	32.39	33.62	30.23
				E → C	14.27	18.28	15.36	13.96	14.11
	√	×	√	C → E	34.59	35.21	35.71	35.56	33.74
	√	√	×	E → C	20.69	25.85	19.76	18.77	19.74
RNNSEARCH	√	×	×	C →E	30.74	35.16	33.75	34.63	31.74
				E →C	15.71	20.76	16.56	16.85	15.14
	√	×	√	C → E	35.61^{**++}	38.78^{**++}	38.32^{**++}	38.49^{**++}	36.45^{**++}
				E →C	17.59^{++}	23.99 $^{++}$	18.95^{++}	18.85^{++}	17.91^{++}
	√	√	×	C →E	35.01^{++}	38.20^{**++}	37.99^{**++}	38.16^{**++}	36.07^{**++}
				E →C	21.12^{*++}	29.52^{**++}	20.49^{**++}	21.59^{**++}	19.97^{++}

Table 2: Comparison with MOSES and RNNSEARCH. MOSES is a phrase-based statistical machine translation system (Koehn et al., 2007). RNNSEARCH is an attention-based neural machine translation system (Bahdanau et al., 2015). "CE" donates Chinese-English parallel corpus, "C" donates Chinese monolingual corpus, and "E" donates English monolingual corpus. "√" means the corpus is included in the training data and × means not included. "NIST06" is the validation set and "NIST02-05" are test sets. The BLEU scores are case-insensitive. "*": significantly better than MOSES ($p < 0.05$); "**": significantly better than MOSES ($p < 0.01$);"+": significantly better than RNNSEARCH ($p < 0.05$); "++": significantly better than RNNSEARCH ($p < 0.01$).

Experiments: Results

Method	Training Data			Direction	NIST06	NIST02	NIST03	NIST04	NIST05
	CE	C	E						
Sennrich et al. (2015)	√	×	√	C →E	34.10	36.95	36.80	37.99	35.33
	√	√	×	E →C	19.85	28.83	20.61	20.54	19.17
this work	√	×	√	C →E	35.61**	38.78**	38.32**	38.49*	36.45**
				E →C	17.59	23.99	18.95	18.85	17.91
	√	√	×	C →E	35.01**	38.20**	37.99**	38.16	36.07**
				E →C	21.12**	29.52**	20.49	21.59**	19.97**

Table 3: Comparison with Sennrich et al. (2015). Both Sennrich et al. (2015) and our approach build on top of RNNSEARCH to exploit monolingual corpora. The BLEU scores are case-insensitive. "*": significantly better than Sennrich et al. (2015) ($p < 0.05$); "**": significantly better than Sennrich et al. (2015) ($p < 0.01$).

Paper 2

Joint Training for Neural Machine Translation Models with Monolingual Data

Zhirui Zhang[†], Shujie Liu[‡], Mu Li[‡], Ming Zhou[‡], Enhong Chen[†*]
[†]University of Science and Technology of China, Hefei, China
[‡]Microsoft Research
[†]zrustc11@gmail.com [†]cheneh@ustc.edu.cn
[‡]{shujliu,muli,mingzhou}@microsoft.com

Model

$$L^*(\theta_{x\to y}) = \sum_{n=1}^{N} \log p(y^{(n)}|x^{(n)}) + \sum_{t=1}^{T} \log p(y^{(t)})$$

$$\log p(y^{(t)}) = \log \sum_x p(x, y^{(t)}) = \log \sum_x Q(x) \frac{p(x, y^{(t)})}{Q(x)}$$

$$\geq \sum_x Q(x) \log \frac{p(x, y^{(t)})}{Q(x)} \text{(Jensen's inequality)}$$

$$= \sum_x [Q(x) \log p(y^{(t)}|x) - KL(Q(x)||p(x))]$$

$$L^*(\theta_{x\to y}) \geq L(\theta_{x\to y}) = \sum_{n=1}^{N} \log p(y^{(n)}|x^{(n)}) +$$

$$\sum_{t=1}^{T} \sum_x [p(x|y^{(t)}) \log p(y^{(t)}|x) - KL(p(x|y^{(t)})||p(x))]$$

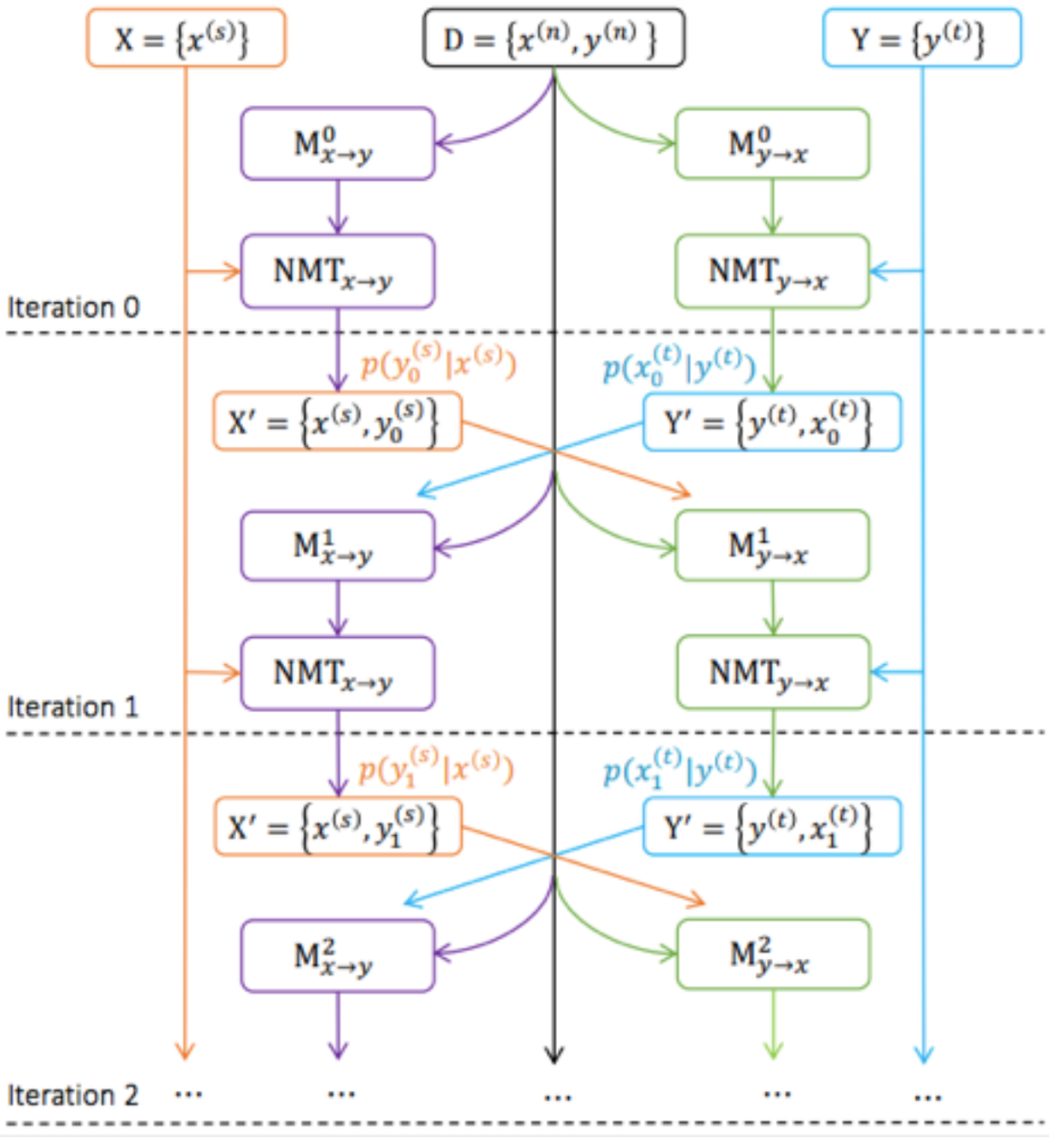
$$L(\theta_{x\to y}) = \sum_{n=1}^{N} \log p(y^{(n)}|x^{(n)})$$

$$+ \sum_{t=1}^{T} \sum_x p(x|y^{(t)}) \log p(y^{(t)}|x)$$

NULL

Model

$$L(\theta) = L(\theta_{x\to y}) + L(\theta_{y\to x})$$



Algorithm 1 Joint Training Algorithm for NMT

procedure PRE-TRAINING
  Initialize $M_{x\to y}$ and $M_{y\to x}$ with random weights $\theta_{x\to y}$ and $\theta_{y\to x}$;
  Pre-train $M_{x\to y}$ and $M_{y\to x}$ on bilingual data $D = \{(x^{(n)}, y^{(n)}\}_{n=1}^{N}$ with Equation 4;
end procedure
procedure JOINT-TRAINING
  while Not Converged **do**
    Use $NMT_{y\to x}$ to generate back-translation x for $Y = \{y^{(t)}\}_{t=1}^{T}$ and build pseudo-parallel corpora $Y' = \{x, y^{(t)}\}_{t=1}^{T}$; ▷ E-Step for $NMT_{x\to y}$
    Use $NMT_{x\to y}$ to generate back-translation y for $X = \{x^{(s)}\}_{s=1}^{S}$ and build pseudo-parallel corpora $X' = \{x^{(s)}, y\}_{s=1}^{S}$; ▷ E-Step for $NMT_{y\to x}$
    Train $M_{x\to y}$ with Equation 10 given weighted bilingual corpora $D \cup Y'$; ▷ M-Step for $NMT_{x\to y}$
    Train $M_{y\to x}$ with Equation 12 given weighted bilingual corpora $D \cup X'$; ▷ M-Step for $NMT_{y\to x}$
  end while
end procedure

Experiments: Setup

- Chinese English dataset
- Labeled dataset: LDC consists of **2.6M** sentence pairs with **65.1M** Chinese words and **67.1M** English words.
- Unlabeled dataset: Chinese (8M) and English (8M) parts of the Xinhua portion of the GIGAWORD corpus as the monolingual corpora.
- Valid dataset: NIST 2006
- Test dataset: NIST 2002, 2003, 2004, 2005
- Vocab: 50K freq

Experiments: Setup

- English German dataset
- Labeled dataset: WMT 14, consists of **4.5M** sentence pairs with **110M** Chinese words and **116M** English words.
- Unlabeled dataset: Chinese (8M) and English (8M) from "News Crawl: articles from 2012" provided by WMT'14.
- Valid dataset: news-test 2012 + news-test 2013
- Test dataset: news-test 2014
- Vocab: 50K freq
- Max sent length: 60
- 50K sub-word tokens as vocabulary based on BPE

Performance: Ch-En

Direction	System	NIST2006	NIST2003	NIST2005	NIST2008	NIST2012	Average
C→E	RNNSearch	38.61	39.39	38.31	30.04	28.48	34.97
	RNNSearch+M	40.66	43.26	41.61	32.48	31.16	37.83
	SS-NMT	41.53	44.03	42.24	33.40	31.58	38.56
	JT-NMT	**42.56**	**45.10**	**44.36**	**34.10**	**32.26**	**39.67**
E→C	RNNSearch	17.75	18.37	17.10	13.14	12.85	15.84
	RNNSearch+M	21.28	21.19	19.53	16.47	15.86	18.87
	SS-NMT	21.62	22.00	19.70	17.06	16.48	19.37
	JT-NMT	**22.56**	**22.98**	**20.95**	**17.62**	**17.39**	**20.30**

Table 1: Case-insensitive BLEU scores (%) on Chinese↔English translation. The "Average" denotes the average BLEU score of all datasets in the same setting. The "C" and "E" denote Chinese and English respectively.

Performance: Ge-En

System	Architecture	E→D	D→E
Jean et al. (2015)	Gated RNN with search + PosUnk	18.97	-
Jean et al. (2015)	Gated RNN with search + PosUnk + 500K vocabs	19.40	-
Shen et al. (2016)	Gated RNN with search + PosUnk + MRT	20.45	-
Luong, Pham, and Manning (2015)	LSTM with 4 layers + dropout + local att. + PosUnk	20.90	-
RNNSearch	Gated RNN with search + BPE	19.78	24.91
RNNSearch+M	Gated RNN with search + BPE + monolingual data	21.89	26.81
SS-NMT	Gated RNN with search + BPE + monolingual data	22.64	27.30
JT-NMT	Gated RNN with search + BPE + monolingual data	**23.60**	**27.98**

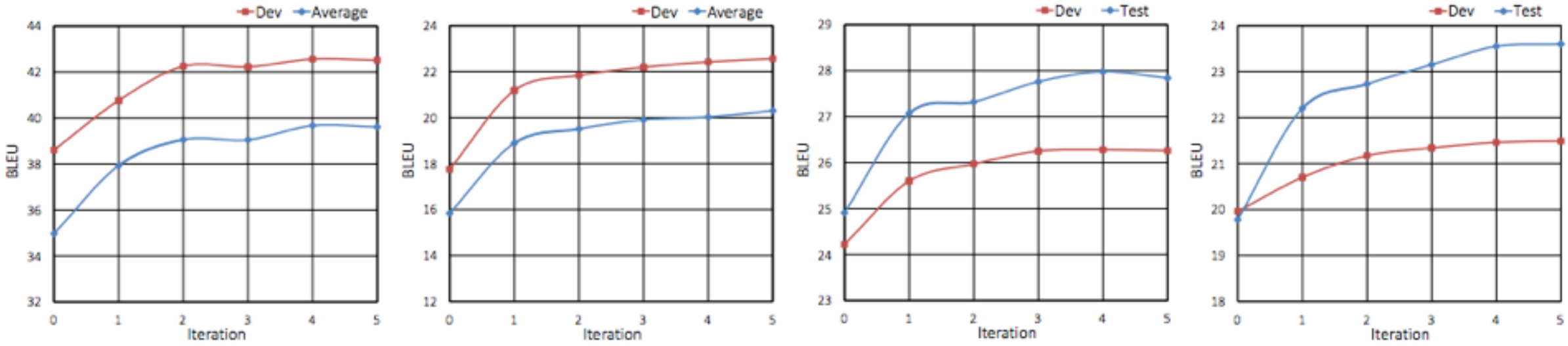
Table 2: Case-sensitive BLEU scores (%) on English↔German translation. "PosUnk" denotes Luong et al. (2015)'s technique of handling rare words. "MRT" denotes minimum risk training proposed in Shen et al. (2016). "BPE" denotes Byte Pair Encoding proposed by Sennrich, Haddow, and Birch (2016b) for word segmentation. The "D" and "E" denote German and English respectively.

Performance: First Iteration

Table 3: The BLEU scores (%) on Chinese↔English and English↔German translation tasks. For Chinese↔English translation, we list the average results of all test sets. For English↔German translation, we list the results of news-test2014.

System	C→E	E→C	D→E	E→D
RNNSearch+M	37.83	18.87	26.81	21.89
JT-NMT (Iteration 1)	**38.23**	**19.10**	**27.07**	**22.20**

Analysis



(a) Chinese-English Translation (b) English-Chinese Translation (c) German-English Translation (d) English-German Translation

Figure 2: BLEU scores (%) on Chinese↔English and English↔German validation and test sets for JT-NMT during training process. "Dev" denotes the results of validation datasets, while "Test" denotes the results of test datasets.

Discussions

- source <-> target are good tasks for semi-sup learn
- Ideas similar to Co-training
- Closely related to unsupervised NMT
- Prior knowledge injection?
- NO idea whether it works in QA tasks