Prior Knowledge Integration for Neural Machine Translation using Posterior Regularization

Jiacheng Zhang[†], **Yang Liu**[†‡*], **Huanbo Luan**[†], **Jingfang Xu**[#] **and Maosong Sun**[†‡]
[†]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
[‡]Jiangsu Collaborative Innovation Center for Language Competence, Jiangsu, China
[#]Sogou Inc., Beijing, China

Abstract

Although neural machine translation has made significant progress recently, how to integrate multiple overlapping, arbitrary prior knowledge sources remains a challenge. In this work, we propose to use posterior regularization to provide a general framework for integrating prior knowledge into neural machine translation. We represent prior knowledge sources as features in a log-linear model, which guides the learning process of the neural translation model. Experiments on Chinese-English translation show that our approach leads to significant improvements.

1 Introduction

The past several years have witnessed the rapid development of neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015), which aims to model the translation process using neural networks in an end-to-end manner. With the capability of capturing long-distance dependencies due to the gating (Hochreiter and Schmidhuber, 1997; Cho et al., 2014) and attention (Bahdanau et al., 2015) mechanisms, NMT has shown remarkable superiority over conventional statistical machine translation (SMT) across a variety of natural languages (Junczys-Dowmunt et al., 2016).

Despite the apparent success, NMT still suffers from one significant drawback: it is difficult to integrate prior knowledge into neural networks. On one hand, neural networks use *continuous* real-valued vectors to represent all language structures involved in the translation process. While these vector representations prove to be capable of capturing translation regularities implicitly (Sutskever et al., 2014), it is hard to interpret each hidden state in neural networks from a linguistic perspective. On the other hand, prior knowledge in machine translation is usually represented in *discrete* symbolic forms such as dictionaries and rules (Nirenburg, 1989) that explicitly encode translation regularities. It is difficult to transform prior knowledge represented in discrete forms to continuous representations required by neural networks.

Therefore, a number of authors have endeavored to integrate prior knowledge into NMT in recent years, either by modifying model architectures (Tu et al., 2016; Cohn et al., 2016; Tang et al., 2016; Feng et al., 2016) or by modifying training objectives (Cohn et al., 2016; Feng et al., 2016; Cheng et al., 2016). For example, to address the over-translation and under-translation problems widely observed in NMT, Tu et al. (2016) directly extend standard NMT to model the coverage constraint that each source phrase should be translated into exactly one target phrase (Koehn et al., 2003). Alternatively, Cohn et al. (2016) and Feng et al. (2016) propose to control the fertilities of source words by appending additional additive terms to training objectives.

Although these approaches have demonstrated clear benefits of incorporating prior knowledge into NMT, how to combine multiple overlapping, arbitrary prior knowledge sources still remains a major challenge. It is difficult to achieve this end by directly modifying model architectures because neural networks usually impose strong independence assumptions between hidden states. As a result, extending a neural model requires that the interdependence of information sources be modeled explicitly (Tu et al., 2016; Tang et al., 2016), making it hard to extend. While this drawback can be partly alleviated by appending additional additive terms to training objectives (Cohn et al., 2016; Feng et al., 2016), these terms are restricted to a

[*] Corresponding author: Yang Liu.

limited number of simple constraints.

In this work, we propose a general framework for integrating multiple overlapping, arbitrary prior knowledge sources into NMT using *posterior regularization* (Ganchev et al., 2010). Our framework is capable of incorporating indirect supervision via posterior distributions of neural translation models. To represent prior knowledge sources as arbitrary real-valued features, we define the posterior distribution as a log-linear model instead of a constrained posterior set (Ganchev et al., 2010). This treatment not only leads to a simpler and more efficient training algorithm but also achieves better translation performance. Experiments show that our approach is able to incorporate a variety of features and achieves significant improvements over posterior regularization using constrained posterior sets on NIST Chinese-English datasets.

2 Background

2.1 Neural Machine Translation

Given a source sentence $\mathbf{x} = x_1, \dots, x_i, \dots, x_I$ and a target sentence $\mathbf{y} = y_1, \dots, y_j, \dots, y_J$, a neural translation model (Sutskever et al., 2014; Bahdanau et al., 2015) is usually factorized as a product of word-level translation probabilities:

$$P(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) = \prod_{j=1}^{J} P(y_j|\mathbf{x}, \mathbf{y}_{<j};\boldsymbol{\theta}), \qquad (1)$$

where $\boldsymbol{\theta}$ is a set of model parameters and $\mathbf{y}_{<j} = y_1, \dots, y_{j-1}$ denotes a partial translation.

The word-level translation probability is defined using a softmax function:

$$P(y_j|\mathbf{x}, \mathbf{y}_{<j};\boldsymbol{\theta}) \propto \exp\Big(f(\mathbf{v}_{y_j}, \mathbf{v}_{\mathbf{x}}, \mathbf{v}_{\mathbf{y}_{<j}}, \boldsymbol{\theta})\Big), \quad (2)$$

where $f(\cdot)$ is a non-linear function, $\mathbf{v}_{y_j}$ is a vector representation of the j-th target word y_j, $\mathbf{v}_{\mathbf{x}}$ is a vector representation of the source sentence $\mathbf{x}$ that encodes the context on the source side, and $\mathbf{v}_{\mathbf{y}_{<j}}$ is a vector representation of the partial translation $\mathbf{y}_{<j}$ that encodes the context on the target side.

Given a training set $\{\langle \mathbf{x}^{(n)}, \mathbf{y}^{(n)} \rangle\}_{n=1}^{N}$, the standard training objective is to maximize the log-likelihood of the training set:

$$\hat{\boldsymbol{\theta}}_{\mathrm{MLE}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \Big\{ \mathcal{L}(\theta) \Big\}, \qquad (3)$$

where

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)};\boldsymbol{\theta}). \qquad (4)$$

Although the introduction of vector representations into machine translation has resulted in substantial improvements in terms of translation quality (Junczys-Dowmunt et al., 2016), it is difficult to incorporate prior knowledge represented in discrete symbolic forms into NMT. For example, given a Chinese-English dictionary containing ground-truth translational equivalents such as ⟨*baigong*, *the White House*⟩, it is non-trivial to leverage the dictionary to guide the learning process of NMT. To address this problem, Tang et al. (2016) propose a new architecture called phraseNet on top of RNNsearch (Bahdanau et al., 2015) that equips standard NMT with an external memory storing phrase tables.

Another important prior knowledge source is the coverage constraint (Koehn et al., 2003): *each source phrase should be translated into exactly one target phrase*. To encode this linguistic intuition into NMT, Tu et al. (2016) extend standard NMT with a coverage vector to keep track of the attention history.

While these approaches are capable of incorporating individual prior knowledge sources separately, how to combine multiple overlapping, arbitrary knowledge sources still remains a major challenge. This can be hardly addressed by modifying model architectures because of the lack of interpretability in NMT and the incapability of neural networks in modeling arbitrary knowledge sources. Although modifying training objectives to include additional knowledge sources as additive terms can partially alleviate this problem, these terms have been restricted to a limited number of simple constraints (Cheng et al., 2016; Cohn et al., 2016; Feng et al., 2016) and incapable of combining arbitrary knowledge sources.

Therefore, it is important to develop a new framework for integrating arbitrary prior knowledge sources into NMT.

2.2 Posterior Regularization

Ganchev et al. (2010) propose *posterior regularization* for incorporating indirect supervision via constraints on posterior distributions of structured latent-variable models. The basic idea is to penalize the log-likelihood of a neural translation model with the KL divergence between a desired distribution that incorporates prior knowledge and the model posteriors. The posterior regularized likelihood is defined as

$$\begin{aligned} & F(\boldsymbol{\theta}, \boldsymbol{q}) \\ = \; & \lambda_1 \mathcal{L}(\boldsymbol{\theta}) - \\ & \lambda_2 \sum_{n=1}^{N} \min_{q \in \mathcal{Q}} \mathrm{KL}\Big(q(\mathbf{y}) \Big|\Big| P(\mathbf{y}|\mathbf{x}^{(n)}; \boldsymbol{\theta}), \Big) \end{aligned} \quad (5)$$

where λ_1 and λ_2 are hyper-parameters to balance the preference between likelihood and posterior regularization, $\mathcal{Q}$ is a set of constrained posteriors:

$$\mathcal{Q} = \{q(\mathbf{y}) : \mathbb{E}_q[\boldsymbol{\phi}(\mathbf{x}, \mathbf{y})] \leq \mathbf{b}\}, \quad (6)$$

where $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$ are constraint features and b is the bound of constraint feature expectations. Ganchev et al. (2010) use constraint features to encode structural bias and define the set of valid distributions with respect to the expectations of constraint features to facilitate inference.

As maximizing $F(\boldsymbol{\theta}, \boldsymbol{q})$ involves minimizing the KL divergence, Ganchev et al. (2010) present a minorization-maximization algorithm akin to EM at sentence level:

$$\mathrm{E} : q^{(t+1)} = \operatorname*{argmin}_{q} \mathrm{KL}\Big(q(\mathbf{y}) \Big|\Big| P(\mathbf{y}|\mathbf{x}^{(n)}; \boldsymbol{\theta}^{(t)})\Big)$$

$$\mathrm{M} : \boldsymbol{\theta}^{(t+1)} = \operatorname*{argmax}_{\boldsymbol{\theta}} \mathbb{E}_{q^{(t+1)}} \Big[\log P(\mathbf{y}|\mathbf{x}^{(n)}; \boldsymbol{\theta})\Big]$$

However, directly applying posterior regularization to neural machine translation faces a major difficulty: it is hard to specify the hyper-parameter $\mathbf{b}$ to effectively bound the expectation of features, which are usually real-valued in translation (Och and Ney, 2002; Koehn et al., 2003; Chiang, 2005). For example, the coverage penalty constraint (Wu et al., 2016) proves to be an essential feature for controlling the length of a translation in NMT. As the value of coverage penalty varies significantly over different sentences, it is difficult to set an appropriate bound for all sentences on the training data. In addition, the minorization-maximization algorithm involves an additional step to find $q^{(t+1)}$ as compared with standard NMT, which increases training time significantly.

3 Posterior Regularization for Neural Machine Translation

3.1 Modeling

In this work, we propose to adapt posterior regularization (Ganchev et al., 2010) to neural machine translation. The major difference is that we represent the desired distribution as a log-linear model (Och and Ney, 2002) rather than a constrained posterior set as described in (Ganchev et al., 2010):

$$\begin{aligned} & \mathcal{J}(\boldsymbol{\theta}, \boldsymbol{\gamma}) \\ & = \lambda_1 \mathcal{L}(\boldsymbol{\theta}) - \\ & \lambda_2 \sum_{n=1}^{N} \mathrm{KL}\Big(Q(\mathbf{y}|\mathbf{x}^{(n)}; \boldsymbol{\gamma}) \Big|\Big| P(\mathbf{y}|\mathbf{x}^{(n)}; \boldsymbol{\theta})\Big), \end{aligned} \quad (7)$$

where the desired distribution that encodes prior knowledge is defined as: [1]

$$Q(\mathbf{y}|\mathbf{x}; \boldsymbol{\gamma}) = \frac{\exp\Big(\boldsymbol{\gamma} \cdot \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})\Big)}{\sum_{\mathbf{y}'} \exp\Big(\boldsymbol{\gamma} \cdot \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}')\Big)}. \quad (8)$$

As compared to previous work on integrating prior knowledge into NMT (Tu et al., 2016; Cohn et al., 2016; Tang et al., 2016), our approach provides a general framework for combining arbitrary knowledge sources. This is due to log-linear models that offer sufficient flexibility to represent arbitrary prior knowledge sources as features. We tackle the representation discrepancy problem by associating the Q distribution that encodes *discrete* representations of prior knowledge with neural models using *continuous* representations learned from data in the KL divergence. Another advantage of our approach is the transparency to model architectures. In principle, our approach can be applied to any neural models for natural language processing.

Our approach also differs from the original version of posterior regularization (Ganchev et al., 2010) in the definition of desired distribution. We resort to log-linear models (Och and Ney, 2002) to incorporate features that have proven effective in SMT. Another benefit of using log-linear models is the differentiability of our training objective (see Eq. (7)). It is easy to leverage standard stochastic gradient descent algorithms to optimize model parameters (Section 3.3).

3.2 Feature Design

In this section, we introduce how to design features to encode prior knowledge in the desired dis-

[1] Ideally, the desired distribution Q should be fixed to guide the learning process of P. However, it is hard to manually specify the feature weights $\boldsymbol{\gamma}$. Therefore, we propose to train both $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ jointly (see Section 3.3). We find that joint training results in significant improvements in practice (see Table 1).

tribution.

Note that not all features in SMT can be adopted to our framework. This is because features in SMT are defined on latent structures such as phrase pairs and synchronous CFG rules, which are not accessible to the decoding process of NMT. Fortunately, we can still leverage internal information in neural models that is linguistically meaningful such as the attention matrix $\mathbf{a}$ (Bahdanau et al., 2015).

We will introduce a number of features used in our experiments as follows.

3.2.1 Bilingual Dictionary

It is natural to leverage a bilingual dictionary $\mathcal{D}$ to improve neural machine translation. Arthur et al. (2016) propose to incorporate discrete translation lexicons into NMT by using the attention vector to select lexical probabilities on which to be focused.

In our work, for each entry $\langle x, y \rangle \in \mathcal{D}$ in the dictionary, a *bilingual dictionary* (BD) feature is defined at the sentence level:

$$\phi_{\mathrm{BD}_{\langle x,y \rangle}}(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 & \text{if } x \in \mathbf{x} \wedge y \in \mathbf{y} \\ 0 & \text{otherwise} \end{cases}. \quad (9)$$

Note that number of bilingual dictionary features depends on the vocabulary of the neural translation model. Entries containing out-of-vocabulary words has to be discarded.

3.2.2 Phrase Table

Phrases, which are sequences of consecutive words, are capable of memorizing local context to deal with word ordering within phrases and translation of short idioms, word insertions or deletions (Koehn et al., 2003; Chiang, 2005). As a result, phrase tables that specify phrase-level correspondences between the source and target languages also prove to be an effective knowledge source in NMT (Tang et al., 2016).

Similar to the bilingual dictionary features, we define a *phrase table* (PT) feature for each entry $\langle \tilde{x}, \tilde{y} \rangle$ in a phrase table $\mathcal{P}$:

$$\phi_{\mathrm{PT}_{\langle \tilde{x},\tilde{y} \rangle}}(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 & \text{if } \tilde{x} \in \mathbf{x} \wedge \tilde{y} \in \mathbf{y} \\ 0 & \text{otherwise} \end{cases}. \quad (10)$$

The number of phrase table features also depends on the vocabulary of the neural translation model.

3.2.3 Coverage Penalty

To overcome the over-translation and under-translation problems widely observed in NMT, a number of authors have proposed to model the fertility (Brown et al., 1993) and converge constraint (Koehn et al., 2003) to improve the adequacy of translation (Tu et al., 2016; Cohn et al., 2016; Feng et al., 2016; Wu et al., 2016; Mi et al., 2016).

We follow Wu et al. (2016) to define a *coverage penalty* (CP) feature to penalize source words with lower sum of attention weights: [2]

$$\phi_{\mathrm{CP}}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{|\mathbf{x}|} \log \Bigg(\min \Big(\sum_{j=1}^{|\mathbf{y}|} \mathbf{a}_{i,j}, 1.0 \Big) \Bigg), \quad (11)$$

where $\mathbf{a}_{i,j}$ is the attention probability of the j-th target word on the i-th source word. Note that the value of coverage penalty feature varies significantly over sentences of different lengths.

3.2.4 Length Ratio

Controlling the length of translations is very important in NMT as neural models tend to generate short translations for long sentences, which deteriorates the translation performance of NMT for long sentences as compared with SMT (Shen et al., 2016).

Therefore, we define the *length ratio* (LR) feature to encourage the length of a translation to fall in a reasonable range:

$$\phi_{\mathrm{LR}}(\mathbf{x}, \mathbf{y}) = \begin{cases} (\beta|\mathbf{x}|)/|\mathbf{y}| & \text{if } \beta|\mathbf{x}| < |\mathbf{y}| \\ |\mathbf{y}|/(\beta|\mathbf{x}|) & \text{otherwise} \end{cases}, \quad (12)$$

where β is a hyper-parameter for penalizing too long or too short translations.

For example, to convey the same meaning, an English sentence is usually about 1.2 times longer than a Chinese sentence. As a result, we can set $\beta = 1.2$. If the length of a Chinese sentence $|\mathbf{x}|$ is 10 and the length of an English sentence $|\mathbf{y}|$ is 12, then, $\phi_{\mathrm{LR}}(\mathbf{x}, \mathbf{y}) = 1$. If the translation is too long (e.g., $|\mathbf{y}| = 100$), then the feature value is 0.12. If the translation is too short (e.g., $|\mathbf{y}| = 6$), the feature value is 0.5.

3.3 Training

In training, our goal is to find a set of model parameters that maximizes the posterior regularized likelihood:

$$\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\theta}, \boldsymbol{\gamma}}{\operatorname{argmax}} \Big\{ \mathcal{J}(\boldsymbol{\theta}, \boldsymbol{\gamma}) \Big\}. \quad (13)$$

[2] For simplicity, we omit the attention matrix $\mathbf{a}$ in the input of the coverage feature function.

Note that unlike the original version of posterior regularization (Ganchev et al., 2010) that relies on a minorization-maximization algorithm to optimize model parameters, our training objective is differentiable with respect to model parameters. Therefore, it is easy to use standard stochastic gradient descent algorithms to train our model.

However, a major difficulty in calculating gradients is that the algorithm needs to sum over all candidate translations in an exponential search space for KL divergence. For example, the partial derivative of $\mathcal{J}(\boldsymbol{\theta}, \gamma)$ with respect to γ is given by

$$\frac{\partial \mathcal{J}(\boldsymbol{\theta}, \gamma)}{\partial \gamma} = -\lambda_2 \times \sum_{n=1}^{N} \frac{\partial}{\partial \gamma} \mathrm{KL}\Big(Q(\mathbf{y}|\mathbf{x}^{(n)}; \gamma)\Big|\Big|P(\mathbf{y}|\mathbf{x}^{(n)}; \boldsymbol{\theta})\Big). \quad (14)$$

The KL divergence is defined as

$$\mathrm{KL}\Big(Q(\mathbf{y}|\mathbf{x}^{(n)}; \gamma)\Big|\Big|P(\mathbf{y}|\mathbf{x}^{(n)}; \boldsymbol{\theta})\Big) = \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}^{(n)})} Q(\mathbf{y}|\mathbf{x}^{(n)}; \gamma) \log \frac{Q(\mathbf{y}|\mathbf{x}^{(n)}; \gamma)}{P(\mathbf{y}|\mathbf{x}^{(n)}; \boldsymbol{\theta})}, \quad (15)$$

where $\mathcal{Y}(\mathbf{x}^{(n)})$ is a set of all possible candidate translations for the source sentence $\mathbf{x}^{(n)}$.

To alleviate this problem, we follow Shen et al. (2016) to approximate the full search space $\mathcal{Y}(\mathbf{x}^{(n)})$ with a sampled sub-space $\mathcal{S}(\mathbf{x}^{(n)})$. Therefore, the KL divergence can be approximated as

$$\mathrm{KL}\Big(Q(\mathbf{y}|\mathbf{x}^{(n)}; \gamma)\Big|\Big|P(\mathbf{y}|\mathbf{x}^{(n)}; \boldsymbol{\theta})\Big) \approx \sum_{\mathbf{y} \in \mathcal{S}(\mathbf{x}^{(n)})} \tilde{Q}(\mathbf{y}|\mathbf{x}^{(n)}; \gamma) \log \frac{\tilde{Q}(\mathbf{y}|\mathbf{x}^{(n)}; \gamma)}{\tilde{P}(\mathbf{y}|\mathbf{x}^{(n)}; \boldsymbol{\theta})}. \quad (16)$$

Note that the Q distribution is also approximated on the sub-space:

$$\tilde{Q}(\mathbf{y}|\mathbf{x}^{(n)}; \gamma) = \frac{\exp(\gamma \cdot \boldsymbol{\phi}(\mathbf{x}^{(n)}, \mathbf{y}))}{\sum_{\mathbf{y}' \in \mathcal{S}(\mathbf{x}^{(n)})} \exp(\gamma \cdot \boldsymbol{\phi}(\mathbf{x}^{(n)}, \mathbf{y}'))}. \quad (17)$$

We follow Shen et al. (2016) to control the sharpness of approximated neural translation distribution normalized on the sampled sub-space:

$$\tilde{P}(\mathbf{y}|\mathbf{x}^{(n)}; \boldsymbol{\theta}) = \frac{P(\mathbf{y}|\mathbf{x}^{(n)}; \boldsymbol{\theta})^{\alpha}}{\sum_{\mathbf{y}' \in \mathcal{S}(\mathbf{x}^{(n)})} P(\mathbf{y}'|\mathbf{x}^{(n)}; \boldsymbol{\theta})^{\alpha}}. \quad (18)$$

3.4 Search

Given learned model parameters $\hat{\boldsymbol{\theta}}$ and $\hat{\gamma}$, the decision rule for translating an unseen source sentence $\mathbf{x}$ is given by

$$\hat{\mathbf{y}} = \underset{\mathcal{Y}(\mathbf{x})}{\mathrm{argmax}} \Big\{ P(\mathbf{y}|\mathbf{x}; \hat{\boldsymbol{\theta}}) \Big\}. \quad (19)$$

The search process can be factorized at the word level:

$$\hat{y}_j = \underset{y \in \mathcal{V}_y}{\mathrm{argmax}} \Big\{ P(y|\mathbf{x}, \hat{\mathbf{y}}_{<j}; \hat{\boldsymbol{\theta}}) \Big\}, \quad (20)$$

where $\mathcal{V}_y$ is the target language vocabulary.

Although this decision rule shares the same efficiency and simplicity with standard NMT (Bahdanau et al., 2015), it does not involve prior knowledge in decoding. Previous studies reveal that incorporating prior knowledge in decoding also significantly boosts translation performance (Arthur et al., 2016; He et al., 2016; Wang et al., 2016).

As directly incorporating prior knowledge into the decoding process of NMT depends on both model structure and the locality of features, we resort to a coarse-to-fine approach instead to keep the architecture transparency of our approach. Given a source sentence $\mathbf{x}$ in the test set, we first use the neural translation model $P(\mathbf{y}|\mathbf{x}; \hat{\boldsymbol{\theta}})$ to generate a k-best list of candidate translation $\mathcal{C}(\mathbf{x})$. Then, the algorithm decides on the most probable candidate translation using the following decision rule:

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{C}(\mathbf{x})}{\mathrm{argmax}} \Big\{ \log P(\mathbf{y}|\mathbf{x}; \hat{\boldsymbol{\theta}}) + \hat{\gamma} \cdot \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) \Big\}. \quad (21)$$

4 Experiments

4.1 Setup

We evaluate our approach on Chinese-English translation. The evaluation metric is case-insensitive BLEU calculated by the *multi-bleu.perl* script. Our training set[3] consists of 1.25M sentence pairs with 27.9M Chinese words and 34.5M English words. We use the NIST 2002 dataset as validation set and the NIST 2003, 2004, 2005, 2006, 2008 datasets as test sets.

In the experiments, we compare our approach with the following two baseline approaches:

[3] The training set includes LDC2002E18, LDC2003E07, LDC2003E14, part of LDC2004T07, LDC2004T08 and LDC2005T06.

Method	Feature	MT02	MT03	MT04	MT05	MT06	MT08	All
RNNSEARCH	N/A	33.45	30.93	32.57	29.86	29.03	21.85	29.11
CPR	N/A	33.84	31.18	33.26	30.67	29.63	22.38	29.72
POSTREG	BD	34.65	31.53	33.82	30.66	29.81	22.55	29.97
	PT	34.56	31.32	33.89	30.70	29.84	22.62	29.99
	LR	34.39	31.41	34.19	30.80	29.82	22.85	30.14
	BD+PT	34.66	32.05	34.54	31.22	30.70	22.84	30.60
	BD+PT+LR	34.37	31.42	34.18	30.99	29.90	22.87	30.20
this work	BD	**36.61**	33.47	36.04	32.96	32.46	**24.78**	32.27
	PT	35.07	32.11	34.73	31.84	30.82	23.23	30.86
	CP	34.68	31.99	34.67	31.37	30.80	23.34	30.76
	LR	34.57	31.89	34.95	31.80	31.43	23.75	31.12
	BD+PT	36.30	**33.83**	36.02	32.98	32.53	24.54	32.29
	BD+PT+CP	36.11	33.64	36.36	**33.11**	32.53	24.57	32.39
	BD+PT+CP+LR	36.10	33.64	**36.48**	33.08	**32.90**	24.63	**32.51**

Table 1: Comparison of BLEU scores on the Chinese-English datasets. RNNSEARCH is an attention-based neural machine translation model (Bahdanau et al., 2015) that does not incorporate prior knowledge. CP extends RNNSEARCH by introducing coverage refinement (Eq. 11) in decoding. POSTREG extends RNNSEARCH with posterior regularization (Ganchev et al., 2010), which uses constraint features to represent prior knowledge and a constrained posterior set to denote the desired distribution. Note that POSTREG cannot use the CP feature (Section 3.2.3) because it is hard to bound the feature value appropriately. On top of RNNSEARCH, our approach also exploits posterior regularization to incorporate prior knowledge but uses a log-linear model to denote the desired distribution. All results of *this work* are significantly better than RNNSEARCH ($p < 0.01$).

1. RNNSEARCH (Bahdanau et al., 2015): a standard attention-based neural machine translation model,

2. CPR (Wu et al., 2016): extending RNNSEARCH by introducing coverage penalty refinement (Eq. (11)) in decoding,

3. POSTREG (Ganchev et al., 2010): extending RNNSEARCH with posterior regularization using constrained posterior set.

For RNNSEARCH, we use an in-house attention-based NMT system that achieves comparable translation performance with GROUNDHOG (Bahdanau et al., 2015), which serves as a baseline approach in our experiments. We limit vocabulary size to 30K for both languages. The word embedding dimension is set to 620. The dimension of hidden layer is set to 1,000. In training, the batch size is set to 80. We use the AdaDelta algorithm (Zeiler, 2012) for optimizing model parameters. In decoding, the beam size is set to 10.

For CPR, we simply follow Wu et al. (2016) to incorporate the coverage penalty into the beam search algorithm of RNNSEARCH.

For POSTREG, we adapt the original version of posterior regularization (Ganchev et al., 2010) to NMT on top of RNNSEARCH. Following Ganchev et al. (2010), we use a ten-step projected gradient descent algorithm to search for an approximate desired distribution in the E step and a one-step gradient descent for the M step.

Our approach extends RNNSEARCH by incorporating prior knowledge. For each source sentence, we sample 80 candidate translations to approximate the $\tilde{P}$ and $\tilde{Q}$ distributions. The hyper-parameter α is set to 0.2. The batch size is 1. The hyper-parameters λ_1 and λ_2 are set to 8×10^{-5} and 2.5×10^{-4}. Note that they not only balance the preference between likelihood and posterior regularization, but also control the values of gradients to fall in a reasonable range for optimization.

We construct bilingual dictionary and phrase table in an automatic way. First, we run the statistical machine translation system MOSES (Koehn and Hoang, 2007) to obtain probabilistic bilingual dictionary and phrase table. For the bilingual dictionary, we retain entries with probabilities higher than 0.1 in both source-to-target and

Feature	Rerank	MT02	MT03	MT04	MT05	MT06	MT08	All
BD	w/o	36.06	32.99	35.62	32.59	32.13	24.36	31.87
	w/	**36.61**	33.47	36.04	32.96	32.46	**24.78**	32.27
PT	w/o	34.98	32.01	34.71	31.77	30.77	23.20	30.81
	w/	35.07	32.11	34.73	31.84	30.82	23.23	30.86
CP	w/o	34.68	31.99	34.67	31.37	30.80	23.34	30.76
	w/	34.68	31.99	34.67	31.37	30.80	23.34	30.76
LR	w/o	34.60	31.89	34.79	31.72	31.39	23.63	31.03
	w/	34.57	31.89	34.95	31.80	31.43	23.75	31.12
BD+PT	w/o	35.76	33.27	35.64	32.47	32.03	24.17	31.83
	w/	36.30	**33.83**	36.02	32.98	32.53	24.54	32.29
BD+PT+CP	w/o	35.71	33.15	35.81	32.52	32.16	24.11	31.89
	w/	36.11	33.64	36.36	**33.11**	32.53	24.57	32.39
BD+PT+CP+LR	w/o	36.06	33.01	35.86	32.70	32.24	24.27	31.96
	w/	36.10	33.64	**36.48**	33.08	**32.90**	24.63	**32.51**

Table 2: Effect of reranking on translation quality.

target-to-source directions. For phrase table, we first remove phrase pairs that occur less than 10 times and then retain entries with probabilities higher than 0.5 in both directions. As a result, both bilingual dictionary and phrase table contain high-quality translation correspondences. We estimate the length ratio on Chinese-English data and set the hyper-parameter β to 1.236.

By default, both PostReg and our approach use reranking to search for the most probable translations (Section 3.4).

4.2 Main Results

Table 1 shows the BLEU scores obtained by RNNSearch, PostReg, and our approach on the Chinese-English datasets.

We find PostReg achieves significant improvements over RNNSearch by adding features that encode prior knowledge. The most effective single feature for PostReg seems to be the length ratio (LR) feature, suggesting that it is important for NMT to control the length of translation to improve translation quality. Note that PostReg is unable to include the coverage penalty (CP) feature because the feature value varies significantly over different sentences. It is hard to specify an appropriate bound $\mathbf{b}$ for constraining the expected feature value. We observe that a loose bound often makes the training process very unstable and fail to converge. Combining features obtains further modest improvements.

Our approach outperforms both RNNSearch and PostReg significantly. The bilingual dictionary (BD) feature turns out to make the most contribution. Compared with CPR that imposes coverage penalty during decoding, our approach that using a single CP feature obtains a significant improvement (i.e., 30.76 over 29.72), suggesting that incorporating prior knowledge sources in modeling might be more beneficial than in decoding.

We find that combining features only results in modest improvements for our approach. One possible reason is that the bilingual dictionary and phrase table features overlap on single word pairs.

4.3 Effect of Reranking

Table 2 shows the effect of reranking on translation quality. We find that using prior knowledge features to rescore the k-best list produced by the neural translation model usually leads to improvements. This finding confirms that adding prior knowledge is beneficial for NMT, either in the training or decoding process.

4.4 Training Speed

Initializing with the best RNNSearch model trained for 300K iterations, our model converges after about 100K iterations. For each iteration, our approach is 1.5 times slower than RNNSearch. When running on a single GPU device Tesla M40, it takes four days to train the RNNSearch model and three extra days to train our model.

4.5 Example Translations

Table 3 gives four examples to demonstrate the benefits of adding features.

Source	*lijing liang tian yu bingxue de* ***fenzhan*** *, 31ri shenye 23 shi 50fen , shanghai jichang jituan* ***yuangong*** ***yinglai*** *le 2004nian de zuihou yige hangban .*
Reference	after **fighting** with ice and snow for two days , **staff** members of shanghai airport group **welcomed** the last flight of 2004 at 23 : 50pm on the 31st .
RNNSEARCH	after a two - day and two - day journey , the team of shanghai 's airport in shanghai has ushered in the last flight in 2004 .
+ BD	after two days and nights **fighting** with ice and snow , the shanghai airport group 's **staff** **welcomed** the last flight in 2004 .
Source	*suiran tonghuopengzhang* ***weilai ji ge yue*** *reng jiang weizhi zai* ***baifenzhier yishang*** *, buguo niandi zhiqian keneng jiangdi .*
Reference	although inflation will remain **above 2 %** **for the coming few months** , it may decline by the end of the year .
RNNSEARCH	although inflation has been maintained for more than two months from the year before the end of the year , it may be lower .
+ PT	although inflation will remain at **more than 2 percent** **in the next few months** , it may be lowered before the end of the year .
Source	*qian ji tian ta ganggang* ***chuyuan*** *, jintian* ***jianchi*** *lai yu lao pengyou* ***daobie*** *.*
Reference	just **discharged from the hospital** a few days ago , he **insisted on** coming to **say farewell** to his old friend today .
RNNSEARCH	during the previous few days , he had just been given treatment to the old friends .
+ CP	during the previous few days , he had just been **discharged from the hospital** , and he **insisted on goodbye** to his old friend today .
Source	*(guoji) yiselie fuzongli fouren jihua kuojian* ***gelan gaodi*** *dingjudian*
Reference	(international) israeli deputy prime minister denied plans to expand **golan heights** settlements
RNNSEARCH	(world) israeli deputy prime minister denies the plan to expand **the golan heights** in **the golan heights**
+ LR	(international) israeli deputy prime minister denies planning to expand **golan heights**

Table 3: Example translations that demonstrate the effect of adding features.

In the first example, source words "fenzhan" (*fighting*), "yuangong" (*staff*), and "yinglai" (*welcomed*) are untranslated in the output of RNNSEARCH. Adding the bilingual dictionary (BD) feature encourages the model to translate these words if they occur in the dictionary.

In the second example, while RNNSEARCH fails to capture phrase cohesion, adding the phrase table (PT) feature is beneficial for translating short idioms, word insertions or deletions that are sensitive to local context.

In the third example, RNNSEARCH tends to omit many source content words such as "chuyuan" (*discharged from the hospital*), "jianchi" (*insisted on*), and "daobie" (*say farewell*). The coverage penalty (CP) feature helps to alleviate the word omission problem.

In the fourth example, the translation produced by RNNSEARCH is too long and "the golan heights" occurs twice. The length ratio (LR) feature is capable of controlling the sentence length in a reasonable range.

5 Related Work

Our work is directly inspired by posterior regularization (Ganchev et al., 2010). The major difference is that we use a log-linear model to represent the desired distribution rather than a constrained posterior set. Using log-linear models not only enables our approach to incorporate arbitrary knowledge sources as real-valued features, but also is differentiable to be jointly trained with neural translation models efficiently.

Our work is closely related to recent work on injecting prior knowledge into NMT (Arthur et al., 2016; Tu et al., 2016; Cohn et al., 2016; Tang et al., 2016; Feng et al., 2016; Wang et al., 2016). The major difference is that our approach aims to provide a general framework for incorporating arbitrary prior knowledge sources while keeping the neural translation model unchanged.

He et al. (2016) also propose to combine the strengths of neural networks on learning representations and log-linear models on encoding prior knowledge. But they treat neural translation models as a feature in the log-linear model. In contrast, we connect the two models via KL divergence to keep the transparency of our approach to model architectures. This enables our approach to be easily applied to other neural models in NLP.

6 Conclusion

We have presented a general framework for incorporating prior knowledge into end-to-end neural machine translation based on posterior regularization (Ganchev et al., 2010). The basic idea is to guide NMT models towards desired behavior using a log-linear model that encodes prior knowledge. Experiments show that incorporating prior knowledge leads to significant improvements over both standard NMT and posterior regularization using constrained posterior sets.

Acknowledgments

We thank Shiqi Shen for useful discussions and anonymous reviewers for insightful comments. This work is supported by the National Natural Science Foundation of China (No.61432013), the 973 Program (2014CB340501), and the National Natural Science Foundation of China (No.61522204). This research is also supported by Sogou Inc. and the Singapore National Research Foundation under its International Research Centre@Singapore Funding Initiative and administered by the IDM Programme.

References

Philip Arthur, Graham Neubug, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. arXiv:1606.02006v2.

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* .

Yong Cheng, Shiqi Shen, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Agreement-based learning of parallel lexicons and phrases from non-parallel corpora. In *Proceedings of IJCAI*.

David Chiang. 2005. A hirarchical phrase-based model for statistical machine translation. In *Proceedings of ACL*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of NAACL*.

Shi Feng, Shujie Liu, Nan Yang, Mu Li, Ming Zhou, and Kenny Q. Zhu. 2016. Improving attention modeling with implicit distortion and fertility for machine translation. In *Proceedings of COLING*.

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research* .

Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. 2016. Improved nerual machine translation with SMT features. In *Proceedings of AAAI*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* .

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is neural machine translation ready for deployment? a case study on 30 translation directions. arXiv:1610.01108v2.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Proceedings of EMNLP*.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation. In *Proceedings of EMNLP*.

Sergei Nirenburg. 1989. Knowledge-based machine translation. *Machine Translation* .

Franz J. Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL*.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of ACL*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*.

Yaohua Tang, Fandong Meng, Zhengdong Lu, Hang Li, and Philip L. H. Yu. 2016. Neural machine translation with external phrase memory. arXiv:1606.01792v1.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of ACL*.

Xing Wang, Zhengdong Lu, Zhaopeng Tu, Hang Li, Deyi Xiong, and Min Zhang. 2016. Neural machine translation advised by statistical machine translation. arXiv:1610.05150.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv:1609.08144v2.

Matthew D. Zeiler. 2012. Adadelta: an adaptive learning rate method. arXiv:1212.5701.