Regular Paper

Unsupervised Word Alignment by Agreement Under ITG Constraint

Hidetaka Kamigaito[1,a)] Akihiro Tamura[2,b)] Hiroya Takamura[3,c)]
Manabu Okumura[3,d)] Eiichiro Sumita[4,e)]

Received: November 28, 2016, Accepted: May 16, 2017

Abstract: We propose a novel unsupervised word alignment method that uses a constraint based on Inversion Transduction Grammar (ITG) parse trees to jointly unify two directional models. Previous agreement methods are not helpful for locating alignments with long distances because they do not use any syntactic structures. In contrast, the proposed method symmetrizes alignments in consideration of their structural coherence by using the ITG constraint softly in the posterior regularization framework. The ITG constraint is also compatible with word alignments that are not covered by ITG parse trees. Hence, the proposed method is robust to ITG parse errors compared to other alignment methods that directly use an ITG model. Compared to the HMM, IBM Model 4, and the baseline agreement method, the experimental results show that, in word alignment evaluation, the IBM Model 4 with the proposed ITG constraint achieves the best performance on the Japanese-English KFTT and BTEC corpus, and in translation evaluation, the proposed method shows comparable or statistically significantly better performance on the Japanese-English KFTT, Japanese-English IWSLT 2007, and Czech/German-English WMT 2015 corpus.

Keywords: statistical machine translation, Inversion Transduction Grammar, unsupervised word alignment, posterior regularized EM, constrained EM

1. Introduction

Word alignment is an important component of statistical machine translation (SMT) systems such as phrase-based SMT [22] and hierarchical phrase-based SMT [7]. In addition, word alignment is utilized for multi-lingual tasks other than SMT, such as bilingual lexicon extraction [25]. The most conventional approaches to word alignment are the IBM models [4] and the HMM model [39], which align each source word to a single target word (i.e., directional models). In these models, bidirectional word alignments are traditionally induced by combining the Viterbi alignments in each direction using heuristics [32]. Matusov et al. [26] exploited a symmetrized posterior probability for bidirectional word alignments. In these methods, each directional model is independently trained.

Previous researches have improved bidirectional word alignments by jointly training two directional models to agree with each other [14], [16], [24]. Such a constraint on the agreement in a training phase is one of the most effective approaches to word alignment. However, none of the previous agreement constraints have taken into account syntactic structures. Therefore, they have difficulty in recovering the alignments with long distances, which frequently occur, especially in grammatically different language pairs.

Some unsupervised word alignment models such as DeNero and Klein [11] and Kondo et al. [23], have been based on syntactic structures. In particular, it has been proven that Inversion Transduction Grammar (ITG) [42], which captures structural coherence between parallel sentences, helps in word alignment [44], [45]. However, ITG has not been introduced into an agreement constraint so far.

We propose an alignment method that uses an ITG constraint to encourage agreement between two directional models in consideration of their structural coherence. Our ITG constraint is based on the Viterbi alignment decided by a bracketing ITG parse tree, and used as a soft constraint in the posterior regularization framework [14]. In addition, our ITG constraint also works on word alignments that are not covered by ITG parse trees, as a standard symmetric constraint. Hence, the proposed method is robust to ITG parse errors compared to an alignment method that uses an ITG directly in model training (e.g., Zhang and Gildea [44], [45]).

Word alignment evaluations show that the proposed ITG constraint achieves significant gains in F-measure and alignment error rate (AER) on the KFTT [30] and the BTEC Japanese-English (Ja-En) corpus [38]. Machine translation evaluations show that our constraint significantly outperforms or is comparable to the baseline symmetric constraint [14] in BLEU on the Ja-En KFTT, Ja-En IWSLT 2007 [12] and Czech (Cs)/German (De)-En WMT 2015 [3] corpus.

[1] NTT Communication Science Laboratories, "Keihanna Science City", Kyoto 619–0237, Japan
[2] Ehime University, Matsuyama, Ehime 790–8577, Japan
[3] Tokyo Institute of Technology, Yokohama, Kanagawa 226–8503, Japan
[4] National Institute of Information and Communications Technology, Soraku-gun, Kyoto 619–0289, Japan
[a)] kamigaito.hidetaka@lab.ntt.co.jp
[b)] tamura@cs.ehime-u.ac.jp
[c)] takamura@pi.titech.ac.jp
[d)] oku@pi.titech.ac.jp
[e)] eiichiro.sumita@nict.go.jp

© 2017 Information Processing Society of Japan

The remainder of this paper is organized as follows. In Section 2, we explain the conventional generative word alignment models, the HMM model and IBM Model 4. In Section 3, we overview the previous agreement method proposed by Ganchev et al. [14]. In Section 4, we propose an unsupervised word alignment method that uses the ITG constraint. In Section 5, we describe the experiments on various language pairs to show the effectiveness of our proposed constraint. In Section 6, we analyze and discuss the results of the experiments. In Section 7, we describe the related work. In Section 8, we summarize this paper.

2. Generative Word Alignment Models

In this section, we present an overview of the conventional generative word alignment models, the HMM model and IBM model 4, into which both previous symmetric and proposed ITG constraints are incorporated. We follow the notation used in Ref. [40].

2.1 HMM Model

The source-to-target HMM model is trained by the EM algorithm as follows.

E-step:

(1) Calculate a source-to-target posterior probability $\overrightarrow{p_\theta}(z_{i,j}|\boldsymbol{x})$ for each bilingual sentence $\boldsymbol{x} = \{\boldsymbol{f}, \boldsymbol{e}\}$, where $\boldsymbol{f} = \{f_1, \cdots, f_I\}$ and $\boldsymbol{e} = \{e_1, \cdots, e_J\}$, under the current model parameters $\overrightarrow{\theta}$ as follows:

$$\overrightarrow{p_\theta}(z_{i,j}|\boldsymbol{x}) = \overrightarrow{\theta}_{align}(a_j|a_{j-1}) \cdot \overrightarrow{\theta}_{lex}(f_i|e_j), \tag{1}$$

where $\overrightarrow{\theta}_{align}(a_j|a_{j-1})$ is the alignment transition probability, $\overrightarrow{\theta}_{lex}(f_i|e_j)$ is the lexical emission probability, and z denotes an alignment in a sentence pair $\boldsymbol{x}$. In particular, $z_{i,j} = 1$, if f_i is aligned to e_j (otherwise $z_{i,j} = 0$).

M-step:

(1) Estimate all parameters $\overrightarrow{\theta}$ (i.e., θ_{align} and θ_{lex}) based on the posterior probability $\overrightarrow{p_\theta}(z_{i,j}|\boldsymbol{x})$ as follows:

$$\begin{aligned}
\overrightarrow{\theta}_{align}(i|i') &\leftarrow \frac{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{p_\theta}(z|\boldsymbol{x}) \cdot n_{i|i'}(\boldsymbol{x}, z)}{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{p_\theta}(z|\boldsymbol{x}) \cdot n_{*|i'}(\boldsymbol{x}, z)} \\
\overrightarrow{\theta}_{lex}(f_i|e_j) &\leftarrow \frac{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{p_\theta}(z|\boldsymbol{x}) \cdot n_{f_i|e_j}(\boldsymbol{x}, z)}{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{p_\theta}(z|\boldsymbol{x}) \cdot n_{*|e_j}(\boldsymbol{x}, z)},
\end{aligned} \tag{2}$$

where $n_{i|i'}(\boldsymbol{x}, z)$ is the number of alignment transition from i' to i in the bilingual sentence x, $n_{*|i'}(\boldsymbol{x}, z)$ is the number of alignment transition from i' to arbitrary location index of the source sentence $\boldsymbol{f}$ in the bilingual sentence pair $\boldsymbol{x}$, $n_{f_i|e_j}(\boldsymbol{x}, z)$ is the number of lexical emmissions from a target word e_j to a source word f_i, and $n_{*|e_j}(\boldsymbol{x}, z)$ is the number of lexical emmission from e_j to an arbitrary source word in the bilingual sentence $\boldsymbol{x}$.

In HMM, marginal numbers for each parameter are efficiently counted by the forward-backward algorithm. The target-to-source HMM model is trained similarly to the above-described source-to-target HMM model.

2.2 IBM Model 4

The source-to-target IBM Model 4 is trained by the EM algorithm as follows.

E-step:

(1) Calculate a source-to-target posterior probability $\overrightarrow{p_\theta}(z_{i,j}|\boldsymbol{x})$ for each bilingual sentence $\boldsymbol{x} = \{\boldsymbol{f}, \boldsymbol{e}\}$, where $\boldsymbol{f} = \{f_1, \cdots, f_I\}$ and $\boldsymbol{e} = \{e_1, \cdots, e_J\}$, under the current model parameters $\overrightarrow{\theta}$ as follows:

$$\begin{aligned}
\overrightarrow{p_\theta}(z_{i,j}|\boldsymbol{x}) \;=\; & p(B_0|B_1 \ldots B_J) \cdot \overrightarrow{\theta}_{fer}(\phi_j|e_j) \\
& \cdot \overrightarrow{\theta}_{head}(B_{i,1} - \overline{B_\rho}|E_{\rho_i}) \\
& \cdot \prod_{k=2}^{\phi_j} \overrightarrow{\theta}_{oth}(B_{j,k} - B_{j,k-1}) \cdot \overrightarrow{\theta}_{lex}(f_i|e_j),
\end{aligned} \tag{3}$$

where B_j is a list of the location index of the source words which are aligned to e_j, $B_{j,k}$ is the index of the k-th source word which is aligned to e_j, B_0 means the set of source words aligned with the empty word, $p(B_0|B_1 \ldots B_J)$ is a distribution over the size of B_0, ϕ_j is a fertility size of the target word e_j, $\overrightarrow{\theta}_{fer}(\phi_j|e_j)$ is a fertility probability, E_j is a word class of a target word e_j, ρ_j is a largest location index of the source words contained in B_j, $\overline{B_j}$ is the average of all elements in B_j, $\overrightarrow{\theta}_{head}(B_{j,1} - \overline{B_{\rho_j}}|E_{\rho_j})$ is a probability model for the first aligned target word (head word), and $\overrightarrow{\theta}_{oth}(B_{j,k} - B_{j,k-1})$ is a probability model for a target word other than the head word (we call such a word "not head word").

M-step:

(1) Estimate all parameters $\overrightarrow{\theta}$ (i.e., $\overrightarrow{\theta}_{fer}$, $\overrightarrow{\theta}_{head}$, $\overrightarrow{\theta}_{oth}$, and $\overrightarrow{\theta}_{lex}$) based on the posterior probability $\overrightarrow{p_\theta}(z_{i,j}|\boldsymbol{x})$ as follows:

$$\begin{aligned}
\overrightarrow{\theta}_{fer}(i|i') &\leftarrow \frac{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{p_\theta}(z|\boldsymbol{x}) \cdot n_{\phi|e}(\boldsymbol{x}, z)}{\sum_{\boldsymbol{x}} \sum_z \sum_{\phi'} \overrightarrow{p_\theta}(z|\boldsymbol{x}) \cdot n_{\phi'|e}(\boldsymbol{x}, z)} \\
\overrightarrow{\theta}_{head}(f_i|e_j) &\leftarrow \frac{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{p_\theta}(z|\boldsymbol{x}) \cdot n^{head}_{\Delta_j|E}(\boldsymbol{x}, z)}{\sum_{\boldsymbol{x}} \sum_z \sum \Delta_{j'} \overrightarrow{p_\theta}(z|\boldsymbol{x}) \cdot n^{head}_{\Delta_{j'}|E}(\boldsymbol{x}, z)} \\
\overrightarrow{\theta}_{oth}(f_i|e_j) &\leftarrow \frac{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{p_\theta}(z|\boldsymbol{x}) \cdot n^{oth}_{\Delta_j}(\boldsymbol{x}, z)}{\sum_{\boldsymbol{x}} \sum_z \sum_{\Delta_{j'}} \overrightarrow{p_\theta}(z|\boldsymbol{x}) \cdot n^{oth}_{\Delta_{j'}}(\boldsymbol{x}, z)} \\
\overrightarrow{\theta}_{lex}(f_i|e_j) &\leftarrow \frac{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{p_\theta}(z|\boldsymbol{x}) \cdot n_{f|e}(\boldsymbol{x}, z)}{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{p_\theta}(z|\boldsymbol{x}) \cdot n_{*|e}(\boldsymbol{x}, z)},
\end{aligned} \tag{4}$$

where $n_{\phi|e}(\boldsymbol{x}, z)$ is the number of fertility ϕ of a target word e, $n^{head}_{\Delta_j|E}$ is the number of a relative location j of a word class of a head word E, and $n^{oth}_{\Delta_j}$ is the number of a relative location j of a word class of a "not head word". In contrast to the HMM model, IBM Model 4 cannot use dynamic programming for counting the marginal numbers of parameters. To solve the problem, the parameters of IBM Model 4 are approximately estimated as in Ref. [32].

The target-to-source IBM Model 4 is trained similarly to the above-described source-to-target IBM Model 4.

3. Previous Agreement Constraint in the Posterior Regularization Framework

This section provides an overview of the previous agreement method proposed by Ganchev et al. [14], which is our baseline. The agreement method trains a bidirectional word alignment model on EM algorithms via the posterior regularization framework where an agreement constraint is used as a soft constraint. This constraint is based on a simple intuition that the alignment probability should be the same regardless of its direction (i.e.,

© 2017 Information Processing Society of Japan

source-to-target or target-to-source). In the posterior regularization framework, the agreement constraint is imposed in the E-step of the EM algorithm. In particular, a bidirectional word alignment model is trained as follows:

E-step:

(1) Calculate a source-to-target posterior probability $\overrightarrow{p_\theta}(z|\boldsymbol{x})$ and a target-to-source posterior probability $\overleftarrow{p_\theta}(z|\boldsymbol{x})$ for each bilingual sentence $\boldsymbol{x} = \{\boldsymbol{f}, \boldsymbol{e}\}$ under the current model parameters $\boldsymbol{\theta} = \{\overrightarrow{\theta}, \overleftarrow{\theta}\}$, where z denotes an alignment in a sentence pair $\boldsymbol{x}$. In particular, $z_{i,j} = 1$, if f_i is aligned to e_j (otherwise $z_{i,j} = 0$).

(a) For HMM Model, each posterior probability is calculated by Eq. (1).

(b) For IBM Model 4, each posterior probability is calculated by Eq. (3).

(2) Symmetrize $\overrightarrow{p_\theta}(z|\boldsymbol{x})$ and $\overleftarrow{p_\theta}(z|\boldsymbol{x})$ under the agreement constraint.

In the posterior regularization framework, $\overrightarrow{p}_\theta(z|\boldsymbol{x})$ and $\overleftarrow{p}_\theta(z|\boldsymbol{x})$ are replaced with $\overrightarrow{q}_\lambda(z|\boldsymbol{x})$ and $\overleftarrow{q}_\lambda(z|\boldsymbol{x})$, defined as follows:

$$\overrightarrow{q}_\lambda(z|\boldsymbol{x}) = \frac{1}{Z_{\overrightarrow{q}}} \cdot \overrightarrow{p}_\theta(z|\boldsymbol{x}) \cdot \exp^{(-\boldsymbol{\lambda}\cdot\boldsymbol{\phi}^{\text{agree}}(\boldsymbol{x},z))},$$

$$Z_{\overrightarrow{q}} = \sum_z \overrightarrow{p}_\theta(z|\boldsymbol{x}) \cdot \exp^{(-\boldsymbol{\lambda}\cdot\boldsymbol{\phi}^{\text{agree}}(\boldsymbol{x},z))},$$

$$\overleftarrow{q}_\lambda(z|\boldsymbol{x}) = \frac{1}{Z_{\overleftarrow{q}}} \cdot \overleftarrow{p}_\theta(z|\boldsymbol{x}) \cdot \exp^{(-\boldsymbol{\lambda}\cdot\boldsymbol{\phi}^{\text{agree}}(\boldsymbol{x},z))},$$

$$Z_{\overleftarrow{q}} = \sum_z \overleftarrow{p}_\theta(z|\boldsymbol{x}) \cdot \exp^{(-\boldsymbol{\lambda}\cdot\boldsymbol{\phi}^{\text{agree}}(\boldsymbol{x},z))},$$

where $Z_{\overrightarrow{q}}$ is a normalization term for $\sum_z \overrightarrow{q}_\lambda(z|\boldsymbol{x}) = 1$ ($Z_{\overleftarrow{q}}$ is analogous) and $\boldsymbol{\lambda}$ is a vector of weight parameters that controls the balance between two directional posterior probabilities. Here, $\boldsymbol{\phi}^{\text{agree}}$ is a feature of the agreement constraint, which assigns each alignment direction to a sign (i.e., +1 or −1). In particular, $\boldsymbol{\phi}^{\text{agree}}$ is defined as follows:

$$\phi^{\text{agree}}_{i,j}(\boldsymbol{x}, z) = \begin{cases} +1 & (z \in \overleftarrow{\boldsymbol{Z}}) \wedge (z_{i,j} = 1), \\ -1 & (z \in \overrightarrow{\boldsymbol{Z}}) \wedge (z_{i,j} = 1), \\ 0 & \text{otherwise}, \end{cases}$$

where $\overrightarrow{\boldsymbol{Z}}$ and $\overleftarrow{\boldsymbol{Z}}$ are sets of possible alignments generated by source-to-target and target-to-source alignment models, respectively.

The agreement constraint is defined as follows:

$$\forall i, \forall j, \overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x}) - \overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x}) = 0, \tag{5}$$

so that $\overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ and $\overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ become equal probabilities for each i, j (i.e., $\overrightarrow{q}_\lambda(z|\boldsymbol{x})$ and $\overleftarrow{q}_\lambda(z|\boldsymbol{x})$ are symmetrical). To satisfy the constraint (5), each $\lambda_{i,j}$ is updated by a stochastic gradient descent. Algorithm 1 shows the update procedure of $\boldsymbol{\lambda}$ on the symmetric constraint. T is an iteration size of stochastic gradient descent [*1].

In particular, based on the symmetric constraint, when $\overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ is larger than $\overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$, $\overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ increases and $\overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ decreases until $\overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ equals $\overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ by increasing $\lambda_{i,j}$. When $\overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ is larger than $\overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$, $\overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ decreases and $\overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ increases until $\overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ equals $\overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ by decreasing $\lambda_{i,j}$.

[*1] We set T to 5 in our experiments.

Algorithm 1 Update of λ on symmetric constraint

```
for i ← 1, ··· , |f| do
    for j ← 1, ··· , |e| do
        λ^0_{i,j} ← 0
for t ← 1, ··· , T do
    for i ← 1, ··· , |f| do
        for j ← 1, ··· , |e| do
            λ^t_{i,j} ← q→_{λ^{t-1}_{i,j}}(z_{i,j}|x) − q←_{λ^{t-1}_{i,j}}(z_{i,j}|x)
```

M-step:

(1) Estimate all parameters $\boldsymbol{\theta}$ based on the symmetrized posterior probabilities $\overrightarrow{q_\lambda}(z|\boldsymbol{x})$ and $\overleftarrow{q_\lambda}(z|\boldsymbol{x})$ as follows:

(a) For HMM Model, the source-to-target parameters, $\overrightarrow{\theta}_{align}$ and $\overrightarrow{\theta}_{lex}$, are estimated as follows:

$$\overrightarrow{\theta}_{align}(i|i') \leftarrow \frac{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{q_\lambda}(z|\boldsymbol{x}) \cdot n_{i|i'}(\boldsymbol{x}, z)}{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{q_\lambda}(z|\boldsymbol{x}) \cdot n_{*|i'}(\boldsymbol{x}, z)}$$

$$\overrightarrow{\theta}_{lex}(f_i|e_j) \leftarrow \frac{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{q_\lambda}(z|\boldsymbol{x}) \cdot n_{f|e}(\boldsymbol{x}, z)}{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{q_\lambda}(z|\boldsymbol{x}) \cdot n_{*|e}(\boldsymbol{x}, z)}. \tag{6}$$

The target-to-source parameters, $\overleftarrow{\theta}_{align}$ and $\overleftarrow{\theta}_{lex}$, are similarly updated.

(b) For IBM Model 4, the source-to-target parameters, $\overrightarrow{\theta}_{fer}$, $\overrightarrow{\theta}_{head}$, $\overrightarrow{\theta}_{oth}$ and $\overrightarrow{\theta}_{lex}$, are estimated as follows:

$$\overrightarrow{\theta}_{fer}(i|i') \leftarrow \frac{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{q_\lambda}(z|\boldsymbol{x}) \cdot n_{\phi|e}(\boldsymbol{x}, z)}{\sum_{\boldsymbol{x}} \sum_z \sum_{\phi'} \overrightarrow{q_\lambda}(z|\boldsymbol{x}) \cdot n_{\phi'|e}(\boldsymbol{x}, z)}$$

$$\overrightarrow{\theta}_{head}(f_i|e_j) \leftarrow \frac{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{q_\lambda}(z|\boldsymbol{x}) \cdot n^{head}_{\Delta_j|E}(\boldsymbol{x}, z)}{\sum_{\boldsymbol{x}} \sum_z \sum \Delta_{j'} \overrightarrow{q_\lambda}(z|\boldsymbol{x}) \cdot n^{head}_{\Delta_{j'}|E}(\boldsymbol{x}, z)}$$

$$\overrightarrow{\theta}_{oth}(f_i|e_j) \leftarrow \frac{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{q_\lambda}(z|\boldsymbol{x}) \cdot n^{oth}_{\Delta_j}(\boldsymbol{x}, z)}{\sum_{\boldsymbol{x}} \sum_z \sum_{\Delta_{j'}} \overrightarrow{q_\lambda}(z|\boldsymbol{x}) \cdot n^{oth}_{\Delta_{j'}}(\boldsymbol{x}, z)}$$

$$\overrightarrow{\theta}_{lex}(f_i|e_j) \leftarrow \frac{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{q_\lambda}(z|\boldsymbol{x}) \cdot n_{f|e}(\boldsymbol{x}, z)}{\sum_{\boldsymbol{x}} \sum_z \overrightarrow{q_\lambda}(z|\boldsymbol{x}) \cdot n_{*|e}(\boldsymbol{x}, z)}, \tag{7}$$

The target-to-source parameters, $\overleftarrow{\theta}_{fer}$, $\overleftarrow{\theta}_{head}$, $\overleftarrow{\theta}_{oth}$, and $\overleftarrow{\theta}_{lex}$, are similarly updated.

4. ITG Constraint in the Posterior Regularization Framework

4.1 Overview

The proposed method introduces an ITG constraint into the posterior regularization framework [14] in model training similar to the previous agreement constraint described in Section 3. The proposed model is trained as follows, where the ITG constraint is imposed in the E-step of the EM algorithm:

E-step:

(1) Calculate a source-to-target posterior probability $\overrightarrow{p_\theta}(z|\boldsymbol{x})$ and a target-to-source posterior probability $\overleftarrow{p_\theta}(z|\boldsymbol{x})$ for each bilingual sentence $\boldsymbol{x} = \{\boldsymbol{f}, \boldsymbol{e}\}$ under the current model parameters $\boldsymbol{\theta}$. This step is the same as the E-step (1) in Section 3.

© 2017 Information Processing Society of Japan

(2) Repeat the following steps for all sentence pairs in the training data.
  (a) Find the Viterbi alignment $\boldsymbol{z}^*$ through ITG parsing (see Section 4.2). Here, $z^*_{i,j} = 1$, if f_i is aligned to e_j (otherwise $z^*_{i,j} = 0$).
  (b) Symmetrize $\overrightarrow{p_\theta}(\boldsymbol{z}|\boldsymbol{x})$ and $\overleftarrow{p_\theta}(\boldsymbol{z}|\boldsymbol{x})$ under the constraint of $\boldsymbol{z}^*$ (see Section 4.3).

M-step:

(1) Estimate all parameters $\boldsymbol{\theta}$ similar to the symmetric constraint, based on the symmetrized posterior probabilities $\overrightarrow{q_\lambda}(\boldsymbol{z}|\boldsymbol{x})$ and $\overleftarrow{q_\lambda}(\boldsymbol{z}|\boldsymbol{x})$ (Section 4.3). This step is the same as the M-step (1) in Section 3.

In contrast to the previous agreement method described in Section 3, the proposed method introduces the ITG parsing step and uses a feature function based on the ITG parse trees. We describe the details of the ITG parsing method and the feature function in Sections 4.2 and 4.3, respectively.

4.2 ITG Parsing

In this section, we present our ITG parsing method, which uses a bracketing ITG [42]. The rules of the bracketing ITG are as follows: $A \rightarrow \langle Y/Z \rangle$, $A \rightarrow [Y/Z]$, $A \rightarrow f_i/e_j$, $A \rightarrow f_i/\epsilon$, and $A \rightarrow \epsilon/e_j$, where A, Y, and Z are non-terminal symbols, f_i and e_j are terminal strings, ϵ is a null symbol, $\langle\rangle$ denotes the inversion of two phrase positions, and [] denotes the reversion of two phrase positions.

In general, a bracketing ITG has $O(|\boldsymbol{f}|^3|\boldsymbol{e}|^3)$ time complexity for parsing a sentence pair $\{\boldsymbol{f}, \boldsymbol{e}\}$, where $|\boldsymbol{f}|$ and $|\boldsymbol{e}|$ are the lengths of $\boldsymbol{f}$ and $\boldsymbol{e}$. For efficient ITG parsing, we use the two-step parsing approach [43], which has been proposed to induce Synchronous Context Free Grammar (SCFG) using K-best pruning [*2] with time complexity $O(|\boldsymbol{f}|^3)$ [*3]. Because ITG is a kind of SCFG, this method can be adopted for our ITG parsing. Algorithm 2 shows the details of our two-step parsing for parsing a bilingual sentence $\boldsymbol{x} = \{\boldsymbol{f}, \boldsymbol{e}\}$. In Algorithm 2, $cube_a$ denotes a list of word alignments, $cube_{ph}$ denotes a list of phrasal rules, and *chart* denotes a hyper-graph, representing connections of all word alignments and phrasal rules. The two-step parsing parses a bilingual sentence in the bottom up manner (Steps 1–34), and then derives the Viterbi alignment $\boldsymbol{z}^*$ in the top down manner from a constructed chart (Steps 35–42).

The bottom up parsing firstly generates lexical rules (Steps 1–21). For the K-best pruning (Step 6), we define the probability for each lexical ITG rule. Note that all K-best pruning for rules in Algorithm 1 are based on their inside probabilities. The probability of a rule $A \rightarrow f_i/e_j$ is defined as:

$$P(A \rightarrow f_i/e_j) = \frac{\overrightarrow{p}_\theta(z_{i,j} = 1|\boldsymbol{x}) + \overleftarrow{p}_\theta(z_{i,j} = 1|\boldsymbol{x})}{2}.$$

We provide a constant value p_{null} [*4] both to $P(A \rightarrow \epsilon/e_j)$ and $P(A \rightarrow f_i/\epsilon)$. In addition, we must provide a probability to a one-to-many alignment because the two step parsing approach must pre-compute probabilities for all one-to-many alignments in the first step (Step 5). One-to-many alignments are composed of $A \rightarrow f_i/e_j$ and $A \rightarrow \epsilon/e_j$ rules. We select a set of rules with the highest probability for a one-to-many alignment using Viterbi algorithm, which has a complexity of $O(|\boldsymbol{e}|)$.

The two step parsing secondly generates phrasal rules (Steps 22–34). To reduce computational cost, the probabilities of phrasal rules $P(A \rightarrow \langle Y/Z \rangle)$ and $P(A \rightarrow [Y/Z])$ are not trained, which are set to 0.5 following Saers et al. (2012) [36].

Algorithm 2 Our two-step parsing

```
    # First step: Bottom up parsing
    # Generate lexical rules
 1: for i ← 1, ··· , |f| do
 2:     for j ← 1, ··· , |e| do
 3:         cube_a ← (A → f_i/e_j)
 4:     cube_a ← (A → f_i/ϵ)
 5:     cube_a ← OneToManyAlignment(i)
 6:     chart[i, i] ← Kbest(cube_a)
 7:     clear cube_a, score_best[i, i], rule_best[i, i]
 8:     for each lexical rule r_lex in chart[i, i] do
 9:         if Score(r_lex) ≥ score_best then
10:             score_best[i, i] ← Score(r_lex)
11:             rule_best[i, i] ← r_lex
12: procedure OneToManyAlignment(i)
13:     clear chart_otm[i, i], r_ret, score_otm
14:     for j ← 1, ··· , |e| do
15:         chart_otm[i, i] ← (A → f_i/e_j)
16:         chart_otm[i, i] ← (A → ϵ/e_j)
17:     for each one-to-many rule r_otm enumerated in chart_otm[i, i] do
18:         if Score(r_otm) ≥ score_otm then
19:             score_otm ← Score(r_otm)
20:             rule_ret ← r_otm
21:     return r_ret
    # Generate phrasal rules
22: for h ← 1, ··· , |f| do
23:     for all i, j s.t j − i = h do
24:         for l ← 1, ··· , h do
25:             for each subspan Y in chart[i, i + h] do
26:                 for each subspan Z in chart[i + h, j] do
27:                     cube_ph ← (A → ⟨Y/Z⟩)
28:                     cube_ph ← (A → [Y/Z])
29:         chart[i, j] ← Kbest(cube_ph)
30:         clear cube_ph, score_best[i, j], rule_best[i, j]
31:         for each phrasal rule r_ph in chart[i, i] do
32:             if Score(r_ph) ≥ score_best then
33:                 score_best[i, j] ← Score(r_ph)
34:                 rule_best[i, j] ← r_ph
    # Second step: Top down viterbi parsing
35: clear z*
36: Backtrack(rule_best[1, |f|])
37: procedure Backtrack(rule_viterbi)
38:     if rule_viterbi is a lexical rule and contain (A → f_i/e_j) then
39:         z*_ij = 1
40:     else
41:         Backtrack(rule_best[span of left child of rule_viterbi])
42:         Backtrack(rule_best[span of right child of rule_viterbi])
```

[*2] We set K to 30 in our experiments.

[*3] Our ITG constarint parses 145 sentences per second in KFTT Japanese-English corpus (sentence lengths are less than 40), and 38 sentences per seconds in Hansard French-English corpus (sentences lengths are less than 80) with 16 threads on Intel(R) Xeon(R) CPU E5-4650 0 @ 2.70 GHz × 2.

[*4] We set p_{null} to 10^{-5}.

© 2017 Information Processing Society of Japan

Table 1 The numbers of parallel sentences for each data set.

Task	Corpus	Language Pair	Train	Dev	Test
Word Alignment	Hansard	Fr-En	1.13M	37	447
	KFTT	Ja-En	330k	653	582
	BTEC	Ja-En	10k	0	10k
Machine Translation	KFTT	Ja-En	330k	1.17k	1.16k
	IWSLT 2007	Ja-En	40k	2.5k	489
	WMT 2015	Cs-En	838k	3k	2.66k
	WMT 2015	De-En	2.16M	3k	2.17k

Algorithm 3 Update of λ on ITG constraint

for $i \leftarrow 1, \cdots, |f|$ **do**
  for $j \leftarrow 1, \cdots, |e|$ **do**
    $\lambda^0_{i,j} \leftarrow 0$
for $t \leftarrow 1, \cdots, T$ **do**
  for $i \leftarrow 1, \cdots, |f|$ **do**
    for $j \leftarrow 1, \cdots, |e|$ **do**
      if $z^*_{i,j} = 1$ **then**
        if $\delta_{i,j}(\boldsymbol{x}, \boldsymbol{z}) < 0$ **then**
          $\lambda^t_{i,j} \leftarrow \overrightarrow{q}_{\lambda^{t-1}_{i,j}}(z_{i,j}|\boldsymbol{x}) - \overleftarrow{q}_0(z_{i,j}|\boldsymbol{x})$
        else if $\delta_{i,j}(\boldsymbol{x}, \boldsymbol{z}) > 0$ **then**
          $\lambda^t_{i,j} \leftarrow \overrightarrow{q}_0(z_{i,j}|\boldsymbol{x}) - \overleftarrow{q}_{\lambda^{t-1}_{i,j}}(z_{i,j}|\boldsymbol{x})$
      else
        $\lambda^t_{i,j} \leftarrow \overrightarrow{q}_{\lambda^{t-1}_{i,j}}(z_{i,j}|\boldsymbol{x}) - \overleftarrow{q}_{\lambda^{t-1}_{i,j}}(z_{i,j}|\boldsymbol{x})$

4.3 Proposed ITG Constraint

This section presents the proposed ITG constraint based on the Viterbi alignment $\boldsymbol{z}^*$, which has previously been identified by the bracketing ITG parsing. The ITG constraint uses a feature $\boldsymbol{\phi}^{\text{ITG}}$ instead of $\boldsymbol{\phi}^{\text{agree}}$:

$$\phi^{\text{ITG}}_{i,j}(\boldsymbol{x}, \boldsymbol{z}) = \begin{cases} 0 & \overleftarrow{Y}(i,j) \wedge (z^*_{i,j} = 1) \wedge (\delta_{i,j}(\boldsymbol{x}, \boldsymbol{z}) < 0), \\ +1 & \overleftarrow{Y}(i,j) \wedge (z^*_{i,j} = 1) \wedge (\delta_{i,j}(\boldsymbol{x}, \boldsymbol{z}) > 0), \\ -1 & \overrightarrow{Y}(i,j) \wedge (z^*_{i,j} = 1) \wedge (\delta_{i,j}(\boldsymbol{x}, \boldsymbol{z}) < 0), \\ 0 & \overrightarrow{Y}(i,j) \wedge (z^*_{i,j} = 1) \wedge (\delta_{i,j}(\boldsymbol{x}, \boldsymbol{z}) > 0), \\ +1 & \overleftarrow{Y}(i,j) \wedge (z^*_{i,j} \neq 1), \\ -1 & \overrightarrow{Y}(i,j) \wedge (z^*_{i,j} \neq 1), \\ 0 & \text{otherwise}, \end{cases}$$

where $\overleftarrow{Y}(i,j) = (\boldsymbol{z} \in \overleftarrow{\boldsymbol{Z}}) \wedge (z_{i,j} = 1)$, $\overrightarrow{Y}(i,j) = (\boldsymbol{z} \in \overrightarrow{\boldsymbol{Z}}) \wedge (z_{i,j} = 1)$, and $\delta_{i,j}(\boldsymbol{x}, \boldsymbol{z}) = \overrightarrow{p}_\theta(z_{i,j} = 1|\boldsymbol{x}) - \overleftarrow{p}_\theta(z_{i,j} = 1|\boldsymbol{x})$. Similarly to $\boldsymbol{\phi}^{\text{agree}}$, $\boldsymbol{\phi}^{\text{ITG}}$ is imposed on $\overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ and $\overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ under the constraint (5). Algorithm 3 shows the update procedure of $\boldsymbol{\lambda}$ on our proposed ITG constraint. If $z^*_{i,j} \neq 1$, our feature $\phi^{\text{ITG}}_{i,j}$ operates similarly to $\phi^{\text{agree}}_{i,j}$ according to the last three rules. If $z^*_{i,j} = 1$, $\boldsymbol{\phi}^{\text{ITG}}$ adjusts probabilities of alignments $\overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ and $\overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ by increasing the lower probability without decreasing the higher probability according to the first four rules.

For example, when $z^*_{i,j} = 1$ and $\overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ is larger than $\overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$, $\overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ is increased until $\overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ equals $\overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ according to the second and fourth rules. When $z^*_{i,j} = 1$ and $\overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ is larger than $\overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$, $\overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ is increased until $\overleftarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ equals $\overrightarrow{q}_{\lambda_{i,j}}(z_{i,j} = 1|\boldsymbol{x})$ according to the first and third rules. As a result, probabilities of word alignments in $\boldsymbol{z}^*$ tend to be higher than those of the other alignments.

5. Evaluation

We compared our proposed ITG constraint (*itg*) with the baseline agreement constraint [14] (*sym*) on word alignment and machine translation tasks. In word alignment evaluations, we used the French-English (Fr-En) Hansard Corpus [28], Ja-En KFTT [*5] [30], and Ja-En BTEC Corpus [38]. We used the first 10K sentence pairs in the training data for the IWSLT 2007 translation task, which were manually annotated with word alignment [8], as the BTEC Corpus. In translation evaluations, we used the Ja-En KFTT, Ja-En IWSLT 2007 translation tasks [*6] and Cs/De-En WMT 2015. In the KFTT and IWSLT 2007 translation tasks, each English language model is trained on target side sentences in the training data. In the WMT 2015 translation task, we used the news-commentary-v11 [*7] and europarl-v7 [*8] parallel datasets for training each translation model, i.e., Cs-En and De-En, and English news-commentary-v11 and europarl-v7 monolingual datasets for training the English language model. Following Williams et al. [41], we used WMT 2014 test data [2] as the development data for Cs-En and De-En language pairs. We conducted compound splitting for all German sentences using compound-splitter.perl in the moses toolkit [*9]. **Table 1** shows each corpus size. In each task, all words in the training and development data were lowercased and tokenized [*10], and sentences with over 80 words on either side were removed from the training data. We used cicada [*11] for training the HMM and IBM Model 4 with our proposed methods.

5.1 Word Alignment Evaluation

We measured the performance of word alignment with Precision, Recall, AER, and F-measure [32]. We used only sure alignments for the evaluation of word alignment performance [13] [*12]. We introduced *itg* and *sym* into the HMM and IBM Model 4. Training is bootstrapped from IBM Model 1, followed by HMM and IBM Model 4. All models were trained with five consecutive

[*5] We used the cleaned dataset distributed on the KFTT official web site (http://www.phontron.com/kftt/index.html).
[*6] BTEC Corpus is a subset of IWSLT 2007. For uniform tokenization, we retokenized all Japanese sentences both in IWSLT 2007 and BTEC Corpus using ChaSen [1].
[*7] http://www.casmacat.eu/corpus/news-commentary.html
[*8] http://www.statmt.org/europarl/
[*9] The moses toolkit can be downloaded from http://www.statmt.org/moses/.
[*10] For Cs, De, and En, we used tokenizer.perl and lowercase.perl in the moses toolkit.
[*11] https://github.com/tarowatanabe/cicada
[*12] Since there exists no distinction for sure-possible alignments in the KFTT and BTEC data sets, we treated all alignments of them as sure alignments.

© 2017 Information Processing Society of Japan

Table 2 Word alignment performance.

	Hansard Fr-En				KFTT Ja-En				BTEC Ja-En			
Method	P	R	F	AER	P	R	F	AER	P	R	F	AER
HMM+*none*	**0.7043**	0.8995	0.7900†	0.0646†	**0.5301**†	0.3971	0.4623	0.5377	**0.6605**	0.3122	0.4425	0.5575
HMM+*sym*	0.7002	**0.9123**†	**0.7923**†	**0.0597**†	0.5156	**0.4281**†	0.4678†	0.5322†	0.5689†	**0.3768**	**0.4534**	**0.5466**
HMM+*itg*	0.6961	0.9049	0.7869	0.0629	0.526	0.4232	**0.4690**	**0.5310**	0.5692	0.3719	0.4499	0.5501
IBM Model 4+*none*	**0.6990**	0.8772	0.7780†	0.0775	**0.7390**	0.4229	0.5379	0.4621	**0.7020**	0.3174	0.4454	0.5546
IBM Model 4+*sym*	0.693†	**0.8910**†	**0.7800**†	**0.0693**†	0.7117	0.4542†	0.5545	0.4455	0.6609	0.3721†	0.4761	0.5239
IBM Model 4+*itg*	0.6957	0.8853	0.7791	0.0710	0.7231	**0.4586**	**0.5613**	**0.4387**	0.6723	**0.3744**	**0.4809**	**0.5191**

Table 3 Machine translation performance.

	KFTT	IWSLT 2007	WMT 2015	
Method	Ja-En	Ja-En	Cs-En	De-En
HMM+*none*	18.9	46.4	11.7	14.3
HMM+*sym*	18.9	46.3	11.9	14.3
HMM+*itg*	**19.2**	**47.0**	**12.0**	**14.6**
IBM Model 4+*none*	18.8	**46.7**†	11.7	14.2
IBM Model 4+*sym*	19.3†	45.9	**11.8**†	**14.3**†
IBM Model 4+*itg*	**19.4**	**46.7**	**11.8**	**14.3**

iterations. In the many-to-many alignment extraction, we used the filtering method [26], where a threshold is optimized on AER of the corresponding baseline model (i.e., HMM+*sym* or IBM Model 4+*sym*) [*13].

Table 2 shows the results of word alignment evaluations, where *none* denotes that the model has no constraint. The values in bold represent the best score, and † indicates that the difference from the corresponding proposed model (i.e., HMM+*itg* or IBM Model 4+*itg*) is not statistically significant according to the paired bootstrap resampling [20] ($p \leq 0.05$). As can be seen in Table 2, on IBM Model 4, *itg* achieved significant improvement against *sym* and *none* in KFTT and BTEC Corpus ($p \leq 0.05$). However, in the Hansard Corpus, *itg* is comparable to *sym*. This indicates that capturing structural coherence by *itg* yields a significant benefit to word alignment in a linguistically different language pair such as Ja-En. On HMM Model, *itg* shows no improvement against *sym* both in KFTT and BTEC Corpus although *itg* achieved significant improvement against *none* both in KFTT and BTEC Corpus. We discuss more details about the effectiveness of the ITG constraint, including a possible reason for this observation, in Section 6.

5.2 Translation Evaluation

We measured translation performance with BLEU [33]. We used the Moses phrase-based SMT systems [21] for decoding. All language models are 5-gram and trained using SRILM [37]. When extracting phrases, we applied the method proposed by Matusov et al. [26], where many-to-many alignments are generated based on the averages of the posterior probabilities from two directional models [*14].

We set the distortion-limit parameter to infinite [*15], and other parameters as default settings. Parameter tuning was conducted by 100-best batch MIRA [5] with 25 iterations.

Table 3 shows the average BLEU of five different tunings. In Table 3, the values in bold represent the best score, and † indicates that the comparisons are not significant over the corresponding proposed model (i.e., HMM+*itg* or IBM Model 4+*itg*) according to the bootstrap resampling test ($p \leq 0.05$). We used multeval [9] for significance testing.

As can be seen in Table 3, in all tasks, *itg* achieved significant improvement against both *none* and *sym* on HMM model. On IBM Model4, *itg* significantly outperforms *none* and is comparable to *sym* in KFTT and WMT 2015, while *itg* significantly outperforms *sym* and is comparable to *none* in IWSLT 2007. We discuss more details about these results in Section 6.

6. Discussion

6.1 Effects of ITG Constraints on Word Alignment and Translation

We discuss the effect of our ITG constraint on word alignment and machine translation. As described in Section 4, the ITG constraint is imposed in the E-step of the EM algorithm, not in decoding steps. In other words, the ITG constraint is not directly applied to the development and test sets. Therefore, for the sentences that are not contained in the training corpus, the word alignments are calculated using the emission, transition, and fertility tables trained with the constraint. This means that the effects of the constraint are implicitly reflected in the alignment results. On the other hand, the effects of the constraint are directly reflected in the machine translation results because the phrase tables are extracted from the posterior probabilities calculated in training steps. We would like to improve our model by imposing our ITG constraint on decoding steps in future.

6.2 Comparison between Symmetric and ITG Constraint

As can be seen in Tables 2 and 3, better word alignment does not always result in better translation, which follows the previous reports [13], [15]. For example, in KFTT, *itg* is comparable to *sym* on IBM Model 4 in machine translation; however, *itg* achieved significant improvement over *sym* in terms of word alignments.

On the other hand, on IBM Model 4 in BTEC, *itg* outperforms *sym* both on word alignment and machine translation. In this section, we try to explore factors of the improvements.

[*13] We tried values from 0.1 to 1.0 at an interval of 0.1.
[*14] The posterior thresholds were decided in the same way as the word alignment evaluation.
[*15] This setting is generally used for Ja-En translation tasks [29].

© 2017 Information Processing Society of Japan

Table 4 Percentage of sentences which have gappy alignments.

Method	KFTT Ja-En	IWSLT2007 Ja-En	WMT15 Cs-En	WMT15 De-En
HMM+*none*	46.63%	1.95%	10.87%	29.34%
HMM+*sym*	49.75%	2.69%	11.47%	31.21%
HMM+*itg*	48.20%	2.62%	10.99%	30.56%
IBM Model 4+*none*	7.45%	1.28%	1.24%	4.23%
IBM Model 4+*sym*	9.88%	2.10%	1.64%	6.06%
IBM Model 4+*itg*	9.18%	1.83%	1.51%	5.97%

Table 5 Word alignment performance for each source sentence length.

(a) Sentences with length from 1 to 10.

	KFTT Ja-En				BTEC Ja-En			
Method	P	R	F	AER	P	R	F	AER
IBM Model 4+*none*	**0.8733**	0.6527	0.7471	0.2529	**0.6990**	0.3671	0.4814	0.5186
IBM Model 4+*sym*	0.8597	0.6802	**0.7595**	**0.2405**	0.6607	0.4363	0.5255	0.4745
IBM Model 4+*itg*	0.8460	**0.6823**	0.7554	0.2446	0.6717	**0.4406**	**0.5321**	**0.4679**

(b) Sentences with length from 11 to 20.

	KFTT Ja-En				BTEC Ja-En			
Method	P	R	F	AER	P	R	F	AER
IBM Model 4+*none*	0.7218	0.4943	0.5868	0.4132	**0.7049**	0.2945	0.4154	0.5846
IBM Model 4+*sym*	0.6939	**0.5419**	0.6086	0.3914	0.6641	0.3446	0.4538	0.5462
IBM Model 4+*itg*	**0.7234**	0.5381	**0.6171**	**0.3829**	0.6760	**0.3452**	**0.4570**	**0.5430**

(c) Sentences with length from 21 to 30.

	KFTT Ja-En				BTEC Ja-En			
Method	P	R	F	AER	P	R	F	AER
IBM Model 4+*none*	**0.7426**	0.4418	0.5540	0.4460	**0.7038**	0.2397	0.3576	0.6424
IBM Model 4+*sym*	0.7118	0.4757	0.5703	0.4297	0.6511	0.2657	0.3773	0.6227
IBM Model 4+*itg*	0.7337	**0.4865**	**0.5850**	**0.4150**	0.6642	**0.2658**	**0.3797**	**0.6203**

(d) Sentence with length more than 30.

	KFTT Ja-En				BTEC Ja-En			
Method	P	R	F	AER	P	R	F	AER
IBM Model 4+*none*	0.7184	0.3805	0.4975	0.5025	**0.7120**	0.1910	0.3012	0.6988
IBM Model 4+*sym*	0.6916	0.4093	0.5143	0.4857	0.6299	0.1971	0.3002	0.6998
IBM Model 4+*itg*	**0.6992**	**0.4127**	**0.5191**	**0.4809**	0.6396	**0.2002**	**0.3050**	**0.6950**

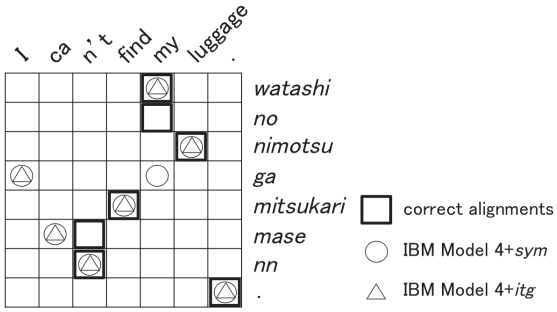


Fig. 1 Word alignment examples on the BTEC corpus.

Figure 1 shows word alignment examples of IBM Model 4+*sym* and IBM Model 4+*itg* on the BTEC corpus. As can be seen in Fig. 1, IBM Model 4+*sym* often generates wrong gappy alignments such as "ga(Ja)-I(En)" and "ga(Ja)-my(En)" while *itg* could prevent such wrong gappy alignments by considering structural coherence. These wrong gappy alignments could disturb the phrase extraction, because excessively long phrase pairs are extracted by bridging the gaps in wrong alignments or simply no phrase pairs are extracted from wrong gappy alignments. **Table 4** shows the percentage of sentences which have gappy alignments in each training corpus used in the translation evaluation. We treat a word alignment bridge with more than phrase extraction limit length [*16] as a gappy alignment. As can be seen in Table 4, gappy alignments generated by *itg* are fewer than those generated by *sym* on all corpora. From the results, a better machine tranlation performance of *itg* than *sym* can be caused by fewer gappy alignments. Note that, from Table 4, *none* is the best model in terms of the number of gappy alignments although translation performance of *none* is lower than that of *sym* and that of *itg*. As reported in Ref. [13], recall of word alignments is more important than precision of word alignments for translation performance. From Table 2, *none* has lower recall of word alignments than *sym* and *itg*. This lower recall might lead to lower translation quality although *none* is superior in terms of gappy alignments.

Some researches such as Ref. [27] have reported that the improvement of word alignment for longer sentences increases the translation qualities. We evaluated the word alignment performance of each IBM Model 4 for each source sentence length on the corpus which is also used in machine translation evaluation, i.e., BTEC and KFTT. The results are shown in **Table 5**, where the values in bold indicate the best scores. Table 5 shows that *itg* outperforms *sym* in AER and F-measure for longer sentences in BTEC and KFTT. In particular, IBM Model4+*itg* is better than

[*16] We used 7 as the phrase extraction limit length in the translation evaluation.

© 2017 Information Processing Society of Japan

IBM Model4+*sym* for sentences with more than 20 in length in KFTT, and IBM Model4+*itg* outperforms IBM Model4+*sym* for all settings in BTEC. From the results, the improvement of the proposed ITG constraint for machine translation performance can be caused by the improvement of word alignment performance for long sentences. In addition, the observation that our ITG constraint is effective for long sentences, supports our expectation that our ITG constraint is helpful for locating alignments with long distances.

6.3 Comparison between HMM and IBM Model 4 with ITG Constraint

In this section, we discuss the difference between the effectiveness of the ITG constraint for HMM and that for IBM Model 4.

In word alignment evaluation, the IBM Model 4+*itg* is better than HMM+*itg* in the Japanese to English language pair. On the other hand, HMM+*itg* is better than IBM Model 4+*itg* in the French to English language pair. In translation evaluation, HMM+*itg* achieves statistically significantly better BLEU scores ($p \leq 0.05$) than IBM Model 4+*itg* on the Czech to English, and the German to English language pairs. In the Japanese to English translation settings, the difference of BLEU scores between HMM+*itg* and IBM Model 4+*itg* is not statistically significant. These observations in word alignment evaluations and machine translation evaluations follow the already-known observations from the comparisons between conventional HMM (i.e., HMM+*none*) and conventional IBM Model 4 (i.e., IBM Model 4+*none*) [40].

Although *itg* achieves the best performance on translation evaluations both for IBM Model 4 and HMM Model (see Table 2), *itg* does not always outperform *sym* on the HMM model in word alignment evaluations. In particular, HMM+*sym* is better than HMM+*itg* on the BTEC corpus. We discuss the reason. In an alignment model with *itg*, word alignments depend on ITG parse trees, and they are updated iteratively through the EM algorithm in the training phase. In early iterations of the EM algorithm, the estimated alignments are less reliable, and ITG parses are conducted based on these less reliable word alignments. These low-quality ITG parse trees can have a bad effect for successive word alignment estimation, and thus lead to the decrease of word alignment performance. We examine the F-measure of the word alignments of the first iteration of the EM estimation for each model in a similar way to the word alignment evaluation in Section 5. **Table 6** shows the results. From Table 6, we can see that the F-measure of HMM+*itg* is significantly lower than that of IBM Model4+*itg* at the first iteration for each corpus. These less reliable initial word alignments can be the reason for the ineffectiveness of the ITG constraint for HMM model.

Table 6 F-measure of the word alignment on first iteration for each model with ITG constraint.

Model	Hansard Fr-En	KFTT Ja-En	BTEC Ja-En
HMM+*itg*	0.7594	0.4272	0.4066
IBM Model 4+*itg*	0.7656	0.5343	0.4800

7. Related Work

Several researches have considered syntactic structures into word alignment models. Riesa et al. [34], [35] have proposed supervised word alignment models based on syntactic structures from constituent parse trees. DeNero and Klein [11] and Kondo et al. [23] have proposed unsupervised word alignment models based on syntactic structures from constituent and dependency parse trees, respectively. However, these models need human-annotated parse trees.

Some alignment models have been based on Inversion Transduction Grammar (ITG) [42], which does not require any human annotations. Haghighi et al. [17] proposed a supervised word alignment method based on ITG structures. Zhang and Gildea [44], [45] used ITG structures for EM estimations, where ITG is used as a hard constraint rather than a soft constraint.

To improve bidirectional word alignments, an agreement constraint has been used. Ganchev et al. [14] have incorporated the symmetric constraint into the posterior regularization framework, a kind of constrained EM. Their method jointly learns both directional two word alignment models. Kamigaito et al. [19] have expanded the Ganchev's agreement constraint to take into account the difference of function words and content words. Note that these methods do not consider syntactic structures.

8. Conclusions

We have proposed a novel unsupervised alignment method that uses an ITG constraint based on bracketing ITG parse trees as a soft constraint of the posterior regularization framework. Due to the ITG constraint, the proposed method can symmetrize two directional alignments based on their structural coherence. Our evaluations have shown that the IBM Model 4 with the proposed ITG constraint achieves the best word alignment performance on the Japanese-English KFTT and BTEC corpus, and significantly improves, or at least keeps, the baseline machine translation performance on the Ja-En KFTT, the Ja-En IWSLT 2007 task, and Cs/De-En WMT 2015. This indicates that the proposed method yields a significant benefit to the machine translation quality, and word alignment quality of linguistically different language pairs.

In future work, we plan to incorporate a phrasal ITG [6], [10], [31] instead of a bracketing ITG to efficiently handle many-to-many alignments.

Acknowledgments This paper is an extension of our manuscript presented by EMNLP 2016 [18] with more detailed analysis and experiments on various language pairs.

References

[1] Asahara, M. and Matsumoto, Y.: Extended models and tools for high-performance part-of-speech tagger, *Proc. 18th Conference on Computational Linguistics-Volume 1*, pp.21–27, Association for Computational Linguistics (2000).

[2] Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L. and Tamchyna, A.: Findings of the 2014 Workshop on Statistical Machine Translation, *Proc. 9th Workshop on Statistical Machine Translation*, pp.12–58, Baltimore, Maryland, USA, Association for Computational Linguistics (2014) (online), available from ⟨http://www.aclweb.org/anthology/W/W14/W14-3302⟩.

[3] Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M.,

© 2017 Information Processing Society of Japan

Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Scarton, C., Specia, L. and Turchi, M.: Findings of the 2015 Workshop on Statistical Machine Translation, *Proc. 10th Workshop on Statistical Machine Translation*, pp.1–46, Lisbon, Portugal, Association for Computational Linguistics (2015) (online), available from ⟨http://aclweb.org/anthology/W15-3001⟩.

[4] Brown, P.F., Pietra, V.J.D., Pietra, S.A.D. and Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation, *Computational Linguistics*, Vol.19, No.2, pp.263–311 (1993).

[5] Cherry, C. and Foster, G.: Batch Tuning Strategies for Statistical Machine Translation, *Proc. 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.427–436, Montréal, Canada, Association for Computational Linguistics (2012) (online), available from ⟨http://www.aclweb.org/anthology/N12-1047⟩.

[6] Cherry, C. and Lin, D.: Inversion Transduction Grammar for Joint Phrasal Translation Modeling, *Proc. SSST, NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation*, pp.17–24, Rochester, New York, Association for Computational Linguistics (2007) (online), available from ⟨http://www.aclweb.org/anthology/W/W07/W07-0403⟩.

[7] Chiang, D.: Hierarchical phrase-based translation, *Computational Linguistics*, Vol.33, No.2, pp.201–228 (2007).

[8] Chooi-Ling, G., Taro, W., Hirofumi, Y. and Eiichiro, S.: Constraining a Generative Word Alignment Model with Discriminative Output (2010).

[9] Clark, J.H., Dyer, C., Lavie, A. and Smith, N.A.: Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability, *Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp.176–181, Portland, Oregon, USA, Association for Computational Linguistics (2011) (online), available from ⟨http://www.aclweb.org/anthology/P11-2031⟩.

[10] Cohn, T. and Haffari, G.: An Infinite Hierarchical Bayesian Model of Phrasal Translation, *Proc. 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp.780–790, Sofia, Bulgaria, Association for Computational Linguistics (2013) (online), available from ⟨http://www.aclweb.org/anthology/P13-1077⟩.

[11] DeNero, J. and Klein, D.: Tailoring Word Alignments to Syntactic Machine Translation, *Proc. 45th Annual Meeting of the Association of Computational Linguistics*, pp.17–24, Prague, Czech Republic, Association for Computational Linguistics (2007) (online), available from ⟨http://www.aclweb.org/anthology/P07-1003⟩.

[12] Fordyce, C.S.: Overview of the IWSLT 2007 evaluation campaign, *Proc. International Workshop on Spoken Language Translation 2007*, pp.1–12 (2007).

[13] Fraser, A. and Marcu, D.: Measuring word alignment quality for statistical machine translation, *Computational Linguistics*, Vol.33, No.3, pp.293–303 (2007).

[14] Ganchev, K., Graca, J., Gillenwater, J. and Taskar, B.: Posterior regularization for structured latent variable models, *The Journal of Machine Learning Research*, Vol.11, pp.2001–2049 (2010).

[15] Ganchev, K., Graça, J.V. and Taskar, B.: Better Alignments = Better Translations?, *Proc. ACL-08: HLT*, pp.986–993, Columbus, Ohio, Association for Computational Linguistics (2008) (online), available from ⟨http://www.aclweb.org/anthology/P/P08/P08-1112⟩.

[16] Graça, J.V., Ganchev, K. and Taskar, B.: Expectation Maximization and Posterior Constraints, *Advances in Neural Information Processing Systems 20*, Platt, J.C., Koller, D., Singer, Y. and Roweis, S.T. (Eds.), pp.569–576, Curran Associates, Inc. (2008) (online), available from ⟨http://papers.nips.cc/paper/3170-expectation-maximization-and-posterior-constraints.pdf⟩.

[17] Haghighi, A., Blitzer, J., DeNero, J. and Klein, D.: Better Word Alignments with Supervised ITG Models, *Proc. Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp.923–931, Suntec, Singapore, Association for Computational Linguistics (2009) (online), available from ⟨http://www.aclweb.org/anthology/P/P09/P09-1104⟩.

[18] Kamigaito, H., Tamura, A., Takamura, H., Okumura, M. and Sumita, E.: Unsupervised Word Alignment by Agreement Under ITG Constraint, *Proc. 2016 Conference on Empirical Methods in Natural Language Processing*, pp.1998–2004, Austin, Texas, Association for Computational Linguistics (2016) (online), available from ⟨https://aclweb.org/anthology/D16-1210⟩.

[19] Kamigaito, H., Watanabe, T., Takamura, H. and Okumura, M.: Unsupervised Word Alignment Using Frequency Constraint in Posterior Regularized EM, *Proc. 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.153–158, Doha, Qatar, Association for Computational Linguistics (2014) (online), available from ⟨http://www.aclweb.org/anthology/D14-1017⟩.

[20] Koehn, P.: Statistical Significance Tests for Machine Translation Evaluation, *Proc. 2004 Conference on Empirical Methods in Natural Language Processing*, Lin, D. and Wu, D. (Eds.), pp.388–395, Barcelona, Spain, Association for Computational Linguistics (2004).

[21] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A. and Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation, *Proc. 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceeding of the Demo and Poster Sessions*, pp.177–180, Prague, Czech Republic, Association for Computational Linguistics (2007) (online), available from ⟨http://www.aclweb.org/anthology/P07-2045⟩.

[22] Koehn, P., Och, F.J. and Marcu, D.: Statistical phrase-based translation, *Proc. 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pp.48–54, Association for Computational Linguistics (2003).

[23] Kondo, S., Duh, K. and Matsumoto, Y.: Hidden Markov Tree Model for Word Alignment, *Proc. 8th Workshop on Statistical Machine Translation*, pp.503–511, Sofia, Bulgaria, Association for Computational Linguistics (2013) (online), available from ⟨http://www.aclweb.org/anthology/W13-2263⟩.

[24] Liang, P., Taskar, B. and Klein, D.: Alignment by Agreement, *Proc. Human Language Technology Conference of the NAACL, Main Conference*, pp.104–111, New York City, USA, Association for Computational Linguistics (2006) (online), available from ⟨http://www.aclweb.org/anthology/N/N06/N06-1014⟩.

[25] Liu, X., Duh, K. and Matsumoto, Y.: Topic Models + Word Alignment = A Flexible Framework for Extracting Bilingual Dictionary from Comparable Corpus, *Proc. 17th Conference on Computational Natural Language Learning*, pp.212–221, Sofia, Bulgaria, Association for Computational Linguistics (2013) (online), available from ⟨http://www.aclweb.org/anthology/W13-3523⟩.

[26] Matusov, E., Zens, R. and Ney, H.: Symmetric Word Alignments for Statistical Machine Translation, *Proc. COLING 2004, the 20th International Conference on Computational Linguistics*, pp.219–225, Geneva, Switzerland, COLING (2004).

[27] Meng, B., Huang, S., Dai, X. and Chen, J.: Segmenting Long Sentence Pairs for Statistical Machine Translation, *2009 International Conference on Asian Language Processing*, pp.53–58 (online), DOI: 10.1109/IALP.2009.20 (2009).

[28] Mihalcea, R. and Pedersen, T.: An Evaluation Exercise for Word Alignment, *Proc. HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, Mihalcea, R. and Pedersen, T. (Eds.), pp.1–10 (2003) (online), available from ⟨http://www.aclweb.org/anthology/W03-0301.pdf⟩.

[29] Murakami, J., Masato, T. and Ikehara, S.: Statistical machine translation using large j/e parallel corpus and long phrase tables, *Proc. International Workshop on Spoken Language Translation 2007*, pp.151–155 (2007).

[30] Neubig, G.: The Kyoto Free Translation Task (2011), available from ⟨http://www.phontron.com/kftt⟩.

[31] Neubig, G., Watanabe, T., Sumita, E., Mori, S. and Kawahara, T.: An Unsupervised Model for Joint Phrase Alignment and Extraction, *Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp.632–641, Portland, Oregon, USA, Association for Computational Linguistics (2011) (online), available from ⟨http://www.aclweb.org/anthology/P11-1064⟩.

[32] Och, F.J. and Ney, H.: A systematic comparison of various statistical alignment models, *Computational Linguistics*, Vol.29, No.1, pp.19–51 (2003).

[33] Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: Bleu: A Method for Automatic Evaluation of Machine Translation, *Proc. 40th Annual Meeting of the Association for Computational Linguistics*, pp.311–318, Philadelphia, Pennsylvania, USA, Association for Computational Linguistics (online), DOI: 10.3115/1073083.1073135 (2002).

[34] Riesa, J., Irvine, A. and Marcu, D.: Feature-Rich Language-Independent Syntax-Based Alignment for Statistical Machine Translation, *Proc. 2011 Conference on Empirical Methods in Natural Language Processing*, pp.497–507, Edinburgh, Scotland, UK., Association for Computational Linguistics (2011) (online), available from ⟨http://www.aclweb.org/anthology/D11-1046⟩.

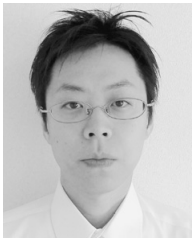
[35] Riesa, J. and Marcu, D.: Hierarchical Search for Word Alignment, *Proc. 48th Annual Meeting of the Association for Computational Linguistics*, pp.157–166, Uppsala, Sweden, Association for Computational Linguistics (2010) (online), available from ⟨http://www.aclweb.org/anthology/P10-1017⟩.

[36] Saers, M., Addanki, K. and Wu, D.: From Finite-State to Inversion Transductions: Toward Unsupervised Bilingual Grammar Induction, *Proc. COLING 2012, the 24th International Conference on Computational Linguistics*, pp.2325–2340, Mumbai, India, The COLING 2012 Organizing Committee (2012) (online), available from

© 2017 Information Processing Society of Japan

⟨http://www.aclweb.org/anthology/C12-1142⟩.
[37] Stolcke, A. et al.: SRILM – An extensible language modeling toolkit, *Proc. International Conference on Spoken Language Processing*, pp.257–286 (2002).
[38] Takezawa, T., Sumita, E., Sugaya, F., Yamamoto, H. and Yamamoto, S.: Toward a Broad-coverage Bilingual Corpus for Speech Translation of Travel Conversations in the Real World, *Proc. 3rd International Conference on Language Resources and Evaluation* (*LREC '02*), pp.147–152 (2002).
[39] Vogel, S., Ney, H. and Tillmann, C.: HMM-based word alignment in statistical translation, *Proc. 16th Conference on Computational Linguistics-Volume 2*, pp.836–841, Association for Computational Linguistics (1996).
[40] Wang, X., Utiyama, M., Finch, A., Watanabe, T. and Sumita, E.: Leave-one-out Word Alignment without Garbage Collector Effects, *Proc. 2015 Conference on Empirical Methods in Natural Language Processing*, pp.1817–1827, Lisbon, Portugal, Association for Computational Linguistics (2015) (online), available from ⟨http://aclweb.org/anthology/D15-1209⟩.
[41] Williams, P., Sennrich, R., Nadejde, M., Huck, M. and Koehn, P.: Edinburgh's Syntax-Based Systems at WMT 2015, *Proc. 10th Workshop on Statistical Machine Translation*, pp.199–209, Lisbon, Portugal, Association for Computational Linguistics (2015) (online), available from ⟨http://aclweb.org/anthology/W15-3024⟩.
[42] Wu, D.: Stochastic inversion transduction grammars and bilingual parsing of parallel corpora, *Computational Linguistics*, Vol.23, No.3, pp.377–403 (1997).
[43] Xiao, X., Xiong, D., Liu, Y., Liu, Q. and Lin, S.: Unsupervised Discriminative Induction of Synchronous Grammar for Machine Translation, *Proc. COLING 2012, the 24th International Conference on Computational Linguistics*, pp.2883–2898, Mumbai, India, The COLING 2012 Organizing Committee (2012) (online), available from ⟨http://www.aclweb.org/anthology/C12-1176⟩.
[44] Zhang, H. and Gildea, D.: Syntax-Based Alignment: Supervised or Unsupervised?, *Proc. COLING 2004, the 20th International Conference on Computational Linguistics*, pp.418–424, Geneva, Switzerland, COLING (2004).
[45] Zhang, H. and Gildea, D.: Stochastic Lexicalized Inversion Transduction Grammar for Alignment, *Proc. 43rd Annual Meeting of the Association for Computational Linguistics* (*ACL '05*), pp.475–482, Ann Arbor, Michigan, Association for Computational Linguistics (online), DOI: 10.3115/1219840.1219899 (2005).

Hidetaka Kamigaito was born in 1989. He received his B.E., M.E. and Dr. Eng. from Tokyo Institute of Technology in 2012, 2014 and 2017, respectively. He is currently a research associate at NTT Communication Science Laboratories. His research interest is machine translation and syntactic parsing.



Akihiro Tamura received his B.E., M.E., and Dr. Eng. from Tokyo Institute of Technology in 2005, 2007, and 2013, respectively. After serving as a researcher at NEC and NICT, he is currently an assistant professor at Ehime University. His research interests include natural language processing and machine learning. He is a member of the Association for Computational Linguistics, IPSJ, the Japanese Society for Artificial Intelligence, and the Association for Natural Language Processing.

Hiroya Takamura received B.E. and M.E. from the University of Tokyo in 1997 and 2000 respectively (in 1999 he was a research student at Technische Universitaet von Wien). He received Dr. Eng. from Nara Institute of Science and Technology in 2003. He was an assistant professor at Tokyo Institute of Technology from 2003 to 2010. He is currently an associate professor at Tokyo Institute of Technology. His current research interest is computational linguistics. He is a member of IPSJ and the Association for Computational Linguistics.

Manabu Okumura was born in 1962. He received B.E., M.E. and Dr. Eng. from Tokyo Institute of Technology in 1984, 1986 and 1989 respectively. He was an assistant at the Department of Computer Science, Tokyo Institute of Technology from 1989 to 1992, and an associate professor at the School of Information Science, Japan Advanced Institute of Science and Technology from 1992 to 2000. He is currently a professor at Institute of Innovative Research, Tokyo Institute of Technology. His current research interests include natural language processing, especially text summarization, computer assisted language learning, sentiment analysis, and text data mining.

Eiichiro Sumita received his Ph.D. in Engineering from Kyoto University in 1999, and a Master's and Bachelor's in Computer Science from the University of Electro-Communications in 1982 and 1980, respectively. He is now in the National Institute of Information and Communication Technology (NICT), its fellow and the associate director-general of Advanced Speech Translation Research and Development Promotion Center (ASTREC). His research interests cover Machine Translation and e-Learning. He is a co-recipient of the Maejima Hisoka Prize in 2013, the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology, Prizes for Science and Technology in 2010, and the AAMT Nagao Award in 2007.

© 2017 Information Processing Society of Japan