Recurrent Neural Networks for Word Alignment Model

Akihiro Tamura[*], **Taro Watanabe, Eiichiro Sumita**
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, JAPAN
a-tamura@ah.jp.nec.com,
{taro.watanabe, eiichiro.sumita}@nict.go.jp

Abstract

This study proposes a word alignment model based on a recurrent neural network (RNN), in which an unlimited alignment history is represented by recurrently connected hidden layers. We perform unsupervised learning using noise-contrastive estimation (Gutmann and Hyvärinen, 2010; Mnih and Teh, 2012), which utilizes artificially generated negative samples. Our alignment model is directional, similar to the generative IBM models (Brown et al., 1993). To overcome this limitation, we encourage agreement between the two directional models by introducing a penalty function that ensures word embedding consistency across two directional models during training. The RNN-based model outperforms the feed-forward neural network-based model (Yang et al., 2013) as well as the IBM Model 4 under Japanese-English and French-English word alignment tasks, and achieves comparable translation performance to those baselines for Japanese-English and Chinese-English translation tasks.

1 Introduction

Automatic word alignment is an important task for statistical machine translation. The most classical approaches are the probabilistic IBM models 1-5 (Brown et al., 1993) and the HMM model (Vogel et al., 1996). Various studies have extended those models. Yang et al. (2013) adapted the Context-Dependent Deep Neural Network for HMM (CD-DNN-HMM) (Dahl et al., 2012), a type of feed-forward neural network (FFNN)-based model, to the HMM alignment model and achieved state-of-the-art performance. However, the FFNN-based model assumes a first-order Markov dependence for alignments.

Recurrent neural network (RNN)-based models have recently demonstrated state-of-the-art performance that outperformed FFNN-based models for various tasks (Mikolov et al., 2010; Mikolov and Zweig, 2012; Auli et al., 2013; Kalchbrenner and Blunsom, 2013; Sundermeyer et al., 2013). An RNN has a hidden layer with recurrent connections that propagates its own previous signals. Through the recurrent architecture, RNN-based models have the inherent property of modeling long-span dependencies, e.g., long contexts, in input data. We assume that this property would fit with a word alignment task, and we propose an RNN-based word alignment model. Our model can maintain and arbitrarily integrate an alignment history, e.g., bilingual context, which is longer than the FFNN-based model.

The NN-based alignment models are supervised models. Unfortunately, it is usually difficult to prepare word-by-word aligned bilingual data. Yang et al. (2013) trained their model from word alignments produced by traditional unsupervised probabilistic models. However, with this approach, errors induced by probabilistic models are learned as correct alignments; thus, generalization capabilities are limited. To solve this problem, we apply noise-contrastive estimation (NCE) (Gutmann and Hyvärinen, 2010; Mnih and Teh, 2012) for unsupervised training of our RNN-based model without gold standard alignments or pseudo-oracle alignments. NCE artificially generates bilingual sentences through samplings as pseudo-negative samples, and then trains the model such that the scores of the original bilingual sentences are higher than those of the sampled bilingual sentences.

Our RNN-based alignment model has a direc-

[*] The first author is now affiliated with Knowledge Discovery Research Laboratories, NEC Corporation, Nara, Japan.

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 1470–1480,
Baltimore, Maryland, USA, June 23-25 2014.

tion, such as other alignment models, i.e., from f (source language) to e (target language) and from e to f. It has been proven that the limitation may be overcome by encouraging two directional models to agree by training them concurrently (Matusov et al., 2004; Liang et al., 2006; Graça et al., 2008; Ganchev et al., 2008). The motivation for this stems from the fact that model and generalization errors by the two models differ, and the models must complement each other. Based on this motivation, our directional models are also simultaneously trained. Specifically, our training encourages word embeddings to be consistent across alignment directions by introducing a penalty term that expresses the difference between embedding of words into an objective function. This constraint prevents each model from overfitting to a particular direction and leads to global optimization across alignment directions.

This paper presents evaluations of Japanese-English and French-English word alignment tasks and Japanese-to-English and Chinese-to-English translation tasks. The results illustrate that our RNN-based model outperforms the FFNN-based model (up to +0.0792 F1-measure) and the IBM Model 4 (up to +0.0703 F1-measure) for the word alignment tasks. For the translation tasks, our model achieves up to 0.74% gain in BLEU as compared to the FFNN-based model, which matches the translation qualities of the IBM Model 4.

2 Related Work

Various word alignment models have been proposed. These models are roughly clustered into two groups: generative models, such as those proposed by Brown et al. (1993), Vogel et al. (1996), and Och and Ney (2003), and discriminative models, such as those proposed by Taskar et al. (2005), Moore (2005), and Blunsom and Cohn (2006).

2.1 Generative Alignment Model

Given a source language sentence $f_1^J = f_1, ..., f_J$ and a target language sentence $e_1^I = e_1, ..., e_I$, f_1^J is generated by e_1^I via the alignment $a_1^J = a_1, ..., a_J$. Each a_j is a hidden variable indicating that the source word f_j is aligned to the target word e_{a_j}. Usually, a "null" word e_0 is added to the target language sentence and a_1^J may contain $a_j = 0$, which indicates that f_j is not aligned to any target word. The probability of generating the sentence f_1^J from e_1^I is defined as

$$p(f_1^J|e_1^I) = \sum_{a_1^J} p(f_1^J, a_1^J|e_1^I). \quad (1)$$

The IBM Models 1 and 2 and the HMM model decompose it into an alignment probability p_a and a lexical translation probability p_t as

$$p(f_1^J, a_1^J|e_1^I) = \prod_{j=1}^{J} p_a(a_j|a_{j-1}, j)p_t(f_j|e_{a_j}). \quad (2)$$

The three models differ in their definition of alignment probability. For example, the HMM model uses an alignment probability with a first-order Markov property: $p_a(a_j|a_j - a_{j-1})$. In addition, the IBM models 3-5 are extensions of these, which consider the fertility and distortion of each translated word.

These models are trained using the expectation-maximization algorithm (Dempster et al., 1977) from bilingual sentences without word-level alignments (unlabeled training data). Given a specific model, the best alignment (Viterbi alignment) of the sentence pair (f_1^J, e_1^I) can be found as

$$\hat{a}_1^J = \underset{a_1^J}{\operatorname{argmax}}\, p(f_1^J, a_1^J|e_1^I). \quad (3)$$

For example, the HMM model identifies the Viterbi alignment using the Viterbi algorithm.

2.2 FFNN-based Alignment Model

As an instance of discriminative models, we describe an FFNN-based word alignment model (Yang et al., 2013), which is our baseline. An FFNN learns a hierarchy of nonlinear features that can automatically capture complex statistical patterns in input data. Recently, FFNNs have been applied successfully to several tasks, such as speech recognition (Dahl et al., 2012), statistical machine translation (Le et al., 2012; Vaswani et al., 2013), and other popular natural language processing tasks (Collobert and Weston, 2008; Collobert et al., 2011).

Yang et al. (2013) have adapted a type of FFNN, i.e., CD-DNN-HMM (Dahl et al., 2012), to the HMM alignment model. Specifically, the lexical translation and alignment probability in Eq. 2 are computed using FFNNs as

$$s_{NN}(a_1^J|f_1^J, e_1^I) = \prod_{j=1}^{J} t_a(a_j - a_{j-1}|c(e_{a_{j-1}})) \cdot t_{lex}(f_j, e_{a_j}|c(f_j), c(e_{a_j})), \quad (4)$$

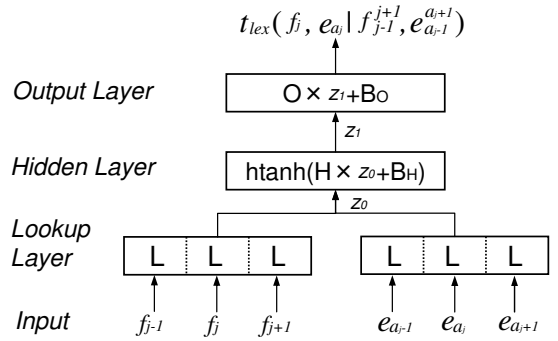

Figure 1: FFNN-based model for computing a lexical translation score of (f_j, e_{a_j})

where t_a and t_{lex} are an alignment score and a lexical translation score, respectively, s_{NN} is a score of alignments a_1^J, and "c(a word w)" denotes a context of word w. Note that the model uses nonprobabilistic scores rather than probabilities because normalization over all words is computationally expensive. The model finds the Viterbi alignment using the Viterbi algorithm, similar to the classic HMM model. Note that alignments in the FFNN-based model are also governed by first-order Markov dynamics because an alignment score depends on the previous alignment a_{j-1}.

Figure 1 shows the network structure with one hidden layer for computing a lexical translation probability $t_{lex}(f_j, e_{a_j}|c(f_j), c(e_{a_j}))$. The model consists of a lookup layer, a hidden layer, and an output layer, which have weight matrices. The model receives a source and target word with their contexts as inputs, which are words in a predefined window (the window size is three in Figure 1). First, the lookup layer converts each input word into its word embedding by looking up its corresponding column in the embedding matrix (L), and then concatenates them. Let V_f (or V_e) be a set of source words (or target words) and M be a predetermined embedding length. L is a $M \times (|V_f| + |V_e|)$ matrix[1]. Word embeddings are dense, low dimensional, and real-valued vectors that can capture syntactic and semantic properties of the words (Bengio et al., 2003). The concatenation (z_0) is then fed to the hidden layer to capture nonlinear relations. Finally, the output layer receives the output of the hidden layer (z_1) and computes a lexical translation score.

[1]We add a special token $\langle unk \rangle$ to handle unknown words and $\langle null \rangle$ to handle null alignments to V_f and V_e

The computations in the hidden and output layer are as follows[2]:

$$z_1 = f(H \times z_0 + B_H), \quad (5)$$

$$t_{lex} = O \times z_1 + B_O, \quad (6)$$

where H, B_H, O, and B_O are $|z_1| \times |z_0|$, $|z_1| \times 1$, $1 \times |z_1|$, and 1×1 matrices, respectively, and $f(x)$ is an activation function. Following Yang et al. (2013), a "hard" version of the hyperbolic tangent, htanh(x)[3], is used as $f(x)$ in this study.

The alignment model based on an FFNN is formed in the same manner as the lexical translation model. Each model is optimized by minimizing the following ranking loss with a margin using stochastic gradient descent (SGD)[4], where gradients are computed by the back-propagation algorithm (Rumelhart et al., 1986):

$$loss(\theta) = \sum_{(\boldsymbol{f},\boldsymbol{e}) \in T} \max\{0, 1 - s_\theta(\boldsymbol{a}^+|\boldsymbol{f}, \boldsymbol{e}) + s_\theta(\boldsymbol{a}^-|\boldsymbol{f}, \boldsymbol{e})\}, \quad (7)$$

where θ denotes the weights of layers in the model, T is a set of training data, $\boldsymbol{a}^+$ is the gold standard alignment, $\boldsymbol{a}^-$ is the incorrect alignment with the highest score under θ, and s_θ denotes the score defined by Eq. 4 as computed by the model under θ.

3 RNN-based Alignment Model

This section proposes an RNN-based alignment model, which computes a score for alignments a_1^J using an RNN:

$$s_{NN}(a_1^J|f_1^J, e_1^I) = \prod_{j=1}^{J} t_{RNN}(a_j|a_1^{j-1}, f_j, e_{a_j}), \quad (8)$$

where t_{RNN} is the score of an alignment a_j. The prediction of the j-th alignment a_j depends on all preceding alignments a_1^{j-1}. Note that the proposed model also uses nonprobabilistic scores, similar to the FFNN-based model.

The RNN-based model is illustrated in Figure 2. The model consists of a lookup layer, a hidden layer, and an output layer, which have weight

[2]Consecutive l hidden layers can be used: $z_l = f(H_l \times z_{l-1} + B_{H_l})$. For simplicity, this paper describes the model with 1 hidden layer.

[3]htanh$(x) = -1$ for $x < -1$, htanh$(x) = 1$ for $x > 1$, and htanh$(x) = x$ for others.

[4]In our experiments, we used a mini-batch SGD instead of a plain SGD.

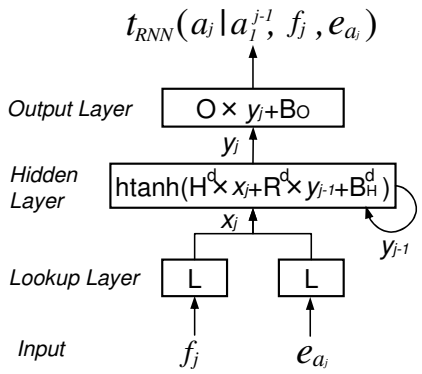

Figure 2: RNN-based alignment model

matrices L, $\{H^d, R^d, B^d_H\}$, and $\{O, B_O\}$, respectively. Each matrix in the hidden layer (H^d, R^d, and B^d_H) depends on alignment, where d denotes the jump distance from a_{j-1} to a_j: $d = a_j - a_{j-1}$. In our experiments, we merge distances that are greater than 8 and less than -8 into the special "$\geq$8" and "$\leq$-8" distances, respectively. Specifically, the hidden layer has weight matrices $\{H^{\leq -8}, H^{-7}, \cdots, H^{7}, H^{\geq 8}, R^{\leq -8}, R^{-7}, \cdots, R^{7}, R^{\geq 8}, B^{\leq -8}_H, B^{-7}_H, \cdots, B^{7}_H, B^{\geq 8}_H\}$ and computes y_j using the corresponding matrices of the jump distance d.

The Viterbi alignment is determined using the Viterbi algorithm, similar to the FFNN-based model, where the model is sequentially applied from f_1 to f_J[5]. When computing the score of the alignment between f_j and e_{a_j}, the two words are input to the lookup layer. In the lookup layer, each of these words is converted to its word embedding, and then the concatenation of the two embeddings (x_j) is fed to the hidden layer in the same manner as the FFNN-based model. Next, the hidden layer receives the output of the lookup layer (x_j) and that of the previous hidden layer (y_{j-1}). The hidden layer then computes and outputs the nonlinear relations between them. Note that the weight matrices used in this computation are embodied by the specific jump distance d. The output of the hidden layer (y_j) is copied and fed to the output layer and the next hidden layer. Finally, the output layer computes the score of a_j ($t_{RNN}(a_j|a_1^{j-1}, f_j, e_{a_j})$) from the output of the hidden layer (y_j). Note that the FFNN-based model consists of two components: one is for lexical translation and the other is for alignment. The proposed RNN produces a single score that is constructed in the hidden layer by employing the distance-dependent weight matrices.

Specifically, the computations in the hidden and output layer are as follows:

$$y_j = f(H^d \times x_j + R^d \times y_{j-1} + B^d_H), \quad (9)$$

$$t_{RNN} = O \times y_j + B_O, \quad (10)$$

where H^d, R^d, B^d_H, O, and B_O are $|y_j| \times |x_j|$, $|y_j| \times |y_{j-1}|$, $|y_j| \times 1$, $1 \times |y_j|$, and 1×1 matrices, respectively. Note that $|y_{j-1}| = |y_j|$. $f(x)$ is an activation function, which is a hard hyperbolic tangent, i.e., htanh(x), in this study.

As described above, the RNN-based model has a hidden layer with recurrent connections. Under the recurrence, the proposed model compactly encodes the entire history of previous alignments in the hidden layer configuration y_i. Therefore, the proposed model can find alignments by taking advantage of the long alignment history, while the FFNN-based model considers only the last alignment.

4 Training

During training, we optimize the weight matrices of each layer (i.e., L, H^d, R^d, B^d_H, O, and B_O) following a given objective using a mini-batch SGD with batch size D, which converges faster than a plain SGD (D = 1). Gradients are computed by the back-propagation through time algorithm (Rumelhart et al., 1986), which unfolds the network in time (j) and computes gradients over time steps. In addition, an $l2$ regularization term is added to the objective to prevent the model from overfitting the training data.

The RNN-based model can be trained by a supervised approach, similar to the FFNN-based model, where training proceeds based on the ranking loss defined by Eq. 7 (Section 2.2). However, this approach requires gold standard alignments. To overcome this drawback, we propose an unsupervised method using NCE, which learns from unlabeled training data.

4.1 Unsupervised Learning

Dyer et al. (2011) presented an unsupervised alignment model based on contrastive estimation (CE) (Smith and Eisner, 2005). CE seeks to discriminate observed data from its neighborhood,

[5]Strictly speaking, we cannot apply the dynamic programming forward-backward algorithm (i.e., the Viterbi algorithm) due to the long alignment history of y_i. Thus, the Viterbi alignment is computed approximately using heuristic beam search.

which can be viewed as pseudo-negative samples. Dyer et al. (2011) regarded all possible alignments of the bilingual sentences, which are given as training data (T), and those of the full translation search space (Ω) as the observed data and its neighborhood, respectively.

We introduce this idea to a ranking loss with margin as

$$loss(\theta) = \max\Bigg\{0, 1 - \sum_{(\boldsymbol{f}^+, \boldsymbol{e}^+) \in T} \mathrm{E}_{\Phi}[s_\theta(\boldsymbol{a}|\boldsymbol{f}^+, \boldsymbol{e}^+)] + \sum_{(\boldsymbol{f}^+, \boldsymbol{e}^-) \in \Omega} \mathrm{E}_{\Phi}[s_\theta(\boldsymbol{a}|\boldsymbol{f}^+, \boldsymbol{e}^-)]\Bigg\}, \quad (11)$$

where Φ is a set of all possible alignments given $(\boldsymbol{f}, \boldsymbol{e})$, $\mathrm{E}_{\Phi}[s_\theta]$ is the expected value of the scores s_θ on Φ, $\boldsymbol{e}^+$ denotes a target language sentence in the training data, and $\boldsymbol{e}^-$ denotes a pseudo-target language sentence. The first expectation term is for the observed data, and the second is for the neighborhood.

However, the computation for Ω is prohibitively expensive. To reduce computation, we employ NCE, which uses randomly sampled sentences from all target language sentences in Ω as $\boldsymbol{e}^-$, and calculate the expected values by a beam search with beam width W to truncate alignments with low scores. In our experiments, we set W to 100. In addition, the above criterion is converted to an online fashion as

$$loss(\theta) = \sum_{\boldsymbol{f}^+ \in T} \max\Bigg\{0, 1 - \mathrm{E}_{\mathrm{GEN}}[s_\theta(\boldsymbol{a}|\boldsymbol{f}^+, \boldsymbol{e}^+)] + \frac{1}{N} \sum_{\boldsymbol{e}^-} \mathrm{E}_{\mathrm{GEN}}[s_\theta(\boldsymbol{a}|\boldsymbol{f}^+, \boldsymbol{e}^-)]\Bigg\}, \quad (12)$$

where $\boldsymbol{e}^+$ is a target language sentence aligned to $\boldsymbol{f}^+$ in the training data, i.e., $(\boldsymbol{f}^+, \boldsymbol{e}^+) \in T$, $\boldsymbol{e}^-$ is a randomly sampled pseudo-target language sentence with length $|\boldsymbol{e}^+|$, and N denotes the number of pseudo-target language sentences per source sentence $\boldsymbol{f}^+$. Note that $|\boldsymbol{e}^+| = |\boldsymbol{e}^-|$. GEN is a subset of all possible word alignments Φ, which is generated by beam search.

In a simple implementation, each $\boldsymbol{e}^-$ is generated by repeating a random sampling from a set of target words (V_e) $|\boldsymbol{e}^+|$ times and lining them up sequentially. To employ more discriminative negative samples, our implementation samples each word of $\boldsymbol{e}^-$ from a set of the target words that co-occur with $f_i \in \boldsymbol{f}^+$ whose probability is above a threshold C under the IBM Model 1 incorporating l_0 prior (Vaswani et al., 2012). The IBM Model 1 with l_0 prior is convenient for reducing translation candidates because it generates more sparse alignments than the standard IBM Model 1.

4.2 Agreement Constraints

Both of the FFNN-based and RNN-based models are based on the HMM alignment model, and they are therefore asymmetric, i.e., they can represent one-to-many relations from the target side. Asymmetric models are usually trained in each alignment direction. The model proposed by Yang et al. (2013) is no exception. However, it has been demonstrated that encouraging directional models to agree improves alignment performance (Matusov et al., 2004; Liang et al., 2006; Graça et al., 2008; Ganchev et al., 2008).

Inspired by their work, we introduce an agreement constraint to our learning. The constraint concretely enforces agreement in word embeddings of both directions. The proposed method trains two directional models concurrently based on the following objective by incorporating a penalty term that expresses the difference between word embeddings:

$$\operatorname*{argmin}_{\theta_{FE}} \big\{loss(\theta_{FE}) + \alpha\|\theta_{L_{EF}} - \theta_{L_{FE}}\|\big\}, \quad (13)$$

$$\operatorname*{argmin}_{\theta_{EF}} \big\{loss(\theta_{EF}) + \alpha\|\theta_{L_{FE}} - \theta_{L_{EF}}\|\big\}, \quad (14)$$

where θ_{FE} (or θ_{EF}) denotes the weights of layers in a source-to-target (or target-to-source) alignment model, θ_L denotes weights of a lookup layer, i.e., word embeddings, and α is a parameter that controls the strength of the agreement constraint. $\|\theta\|$ indicates the norm of θ. 2-norm is used in our experiments. Equations 13 and 14 can be applied to both supervised and unsupervised approaches. Equations 7 and 12 are substituted into $loss(\theta)$ in supervised and unsupervised learning, respectively. The proposed constraint penalizes overfitting to a particular direction and enables two directional models to optimize across alignment directions globally.

Our unsupervised learning procedure is summarized in Algorithm 1. In Algorithm 1, line 2 randomly samples D bilingual sentences $(\mathbf{f}^+, \mathbf{e}^+)^D$ from training data T. Lines 3-1 and 3-2 generate N pseudo-negative samples for each $\mathbf{f}^+$ and $\mathbf{e}^+$ based on the translation candidates of $\mathbf{f}^+$ and $\mathbf{e}^+$ found by the IBM Model 1 with l_0 prior,

Algorithm 1 Training Algorithm

Input: θ^1_{FE}, θ^1_{EF}, training data T, $MaxIter$, batch size D, N, C, $IBM1$, W, α

1: **for all** t such that $1 \leq t \leq MaxIter$ **do**

2: $\{(\mathbf{f}^+, \mathbf{e}^+)^D\} \leftarrow \text{sample}(D, T)$

3-1: $\{(\mathbf{f}^+, \{\mathbf{e}^-\}^N)^D\} \leftarrow \text{neg}_\text{e}(\{(\mathbf{f}^+, \mathbf{e}^+)^D\}, N, C, IBM1)$

3-2: $\{(\mathbf{e}^+, \{\mathbf{f}^-\}^N)^D\} \leftarrow \text{neg}_\text{f}(\{(\mathbf{f}^+, \mathbf{e}^+)^D\}, N, C, IBM1)$

4-1: $\theta^{t+1}_{FE} \leftarrow \text{update}((\mathbf{f}^+, \mathbf{e}^+, \{\mathbf{e}^-\}^N)^D, \theta^t_{FE}, \theta^t_{EF}, W, \alpha)$

4-2: $\theta^{t+1}_{EF} \leftarrow \text{update}((\mathbf{e}^+, \mathbf{f}^+, \{\mathbf{f}^-\}^N)^D, \theta^t_{EF}, \theta^t_{FE}, W, \alpha)$

5: **end for**

Output: $\theta^{MaxIter+1}_{EF}, \theta^{MaxIter+1}_{FE}$

		Train	Dev	Test
$BTEC$		9 K	0	960
$Hansards$		1.1 M	37	447
$FBIS$	$NIST03$	240 K	878	919
	$NIST04$			1,597
$IWSLT$		40 K	2,501	489
$NTCIR$		3.2 M	2,000	2,000

Table 1: Size of experimental datasets

$IBM1$ (Section 4.1). Lines 4-1 and 4-2 update the weights in each layer following a given objective (Sections 4.1 and 4.2). Note that θ^t_{FE} and θ^t_{EF} are concurrently updated in each iteration, and θ^t_{EF} (or θ^t_{FE}) is employed to enforce agreement between word embeddings when updating θ^t_{FE} (or θ^t_{EF}).

5 Experiment

5.1 Experimental Data

We evaluated the alignment performance of the proposed models with two tasks: Japanese-English word alignment with the Basic Travel Expression Corpus ($BTEC$) (Takezawa et al., 2002) and French-English word alignment with the Hansard dataset ($Hansards$) from the 2003 NAACL shared task (Mihalcea and Pedersen, 2003). In addition, we evaluated the end-to-end translation performance of three tasks: a Chinese-to-English translation task with the FBIS corpus ($FBIS$), the IWSLT 2007 Japanese-to-English translation task ($IWSLT$) (Fordyce, 2007), and the NTCIR-9 Japanese-to-English patent translation task ($NTCIR$) (Goto et al., 2011)[6].

Table 1 shows the sizes of our experimental datasets. Note that the development data was not used in the alignment tasks, i.e., $BTEC$ and $Hansards$, because the hyperparameters of the alignment models were set by preliminary small-scale experiments. The $BTEC$ data is the first 9,960 sentence pairs in the training data for $IWSLT$, which were annotated with word alignment (Goh et al., 2010). We split these pairs into the first 9,000 for training data and the remaining 960 as test data. All the data in $BTEC$ is word-aligned, and the training data in $Hansards$ is unlabeled data. In $FBIS$, we used the NIST02 evaluation data as the development data, and the NIST03 and 04 evaluation data as test data ($NIST03$ and $NIST04$).

[6]We did not evaluate the translation performance on the Hansards data because the development data is very small and performance is unreliable.

5.2 Comparing Methods

We evaluated the proposed RNN-based alignment models against two baselines: the IBM Model 4 and the FFNN-based model with one hidden layer. The IBM Model 4 was trained by previously presented model sequence schemes (Och and Ney, 2003): $1^5H^53^54^5$, i.e., five iterations of the IBM Model 1 followed by five iterations of the HMM Model, etc., which is the default setting for GIZA++ ($IBM4$). For the FFNN-based model, we set the word embedding length M to 30, the number of units of a hidden layer $|z_1|$ to 100, and the window size of contexts to 5. Hence, $|z_0|$ is 300 ($30 \times 5 \times 2$). Following Yang et al. (2013), the FFNN-based model was trained by the supervised approach described in Section 2.2 ($FFNN_s$).

For the RNN-based models, we set M to 30 and the number of units of each recurrent hidden layer $|y_j|$ to 100. Thus, $|x_j|$ is 60 (30×2). The number of units of each layer of the FFNN-based and RNN-based models and M were set through preliminary experiments. To demonstrate the effectiveness of the proposed learning methods, we evaluated four types of RNN-based models: RNN_s, RNN_{s+c}, RNN_u, and RNN_{u+c}, where "*s*/*u*" denotes a supervised/unsupervised model and "$+c$" indicates that the agreement constraint was used.

In training all the models except $IBM4$, the weights of each layer were initialized first. For the weights of a lookup layer L, we preliminarily trained word embeddings for the source and target language from each side of the training data. We then set the word embeddings to L to avoid falling into local minima. Other weights were randomly initialized to $[-0.1, 0.1]$. For the pretraining, we

Alignment	$BTEC$	$Hansards$
$IBM4$	0.4859	0.9029
$FFNN_s$(I)	0.4770	0.9020
RNN_s(I)	0.5053$^+$	0.9068
RNN_{s+c}(I)	0.5174$^+$	0.9202$^+$
RNN_u	0.5307$^+$	0.9037
RNN_{u+c}	**0.5562**$^+$	**0.9275**$^+$
$FFNN_s$(R)	0.8224	-
RNN_s(R)	0.8798$^+$	-
RNN_{s+c}(R)	**0.8921**$^+$	-

Table 2: Word alignment performance (F1-measure)

used the RNNLM Toolkit [7] (Mikolov et al., 2010) with the default options. We mapped all words that occurred less than five times to the special token $\langle unk \rangle$. Next, each weight was optimized using the mini-batch SGD, where batch size D was 100, learning rate was 0.01, and an l_2 regularization parameter was 0.1. The training stopped after 50 epochs. The other parameters were set as follows: W, N and C in the unsupervised learning were 100, 50, and 0.001, respectively, and α for the agreement constraint was 0.1.

In the translation tasks, we used the Moses phrase-based SMT systems (Koehn et al., 2007). All Japanese and Chinese sentences were segmented by ChaSen[8] and the Stanford Chinese segmenter[9], respectively. In the training, long sentences with over 40 words were filtered out. Using the SRILM Toolkits (Stolcke, 2002) with modified Kneser-Ney smoothing, we trained a 5-gram language model on the English side of each training data for $IWSLT$ and $NTCIR$, and a 5-gram language model on the Xinhua portion of the English Gigaword corpus for $FBIS$. The SMT weighting parameters were tuned by MERT (Och, 2003) in the development data.

5.3 Word Alignment Results

Table 2 shows the alignment performance by the F1-measure. Hereafter, $MODEL(R)$ and $MODEL(I)$ denote the $MODEL$ trained from gold standard alignments and word alignments found by the IBM Model 4, respectively. In $Hansards$, all models were trained from randomly sampled 100 K data[10]. We evaluated the word alignments produced by first applying each model in both directions and then combining the alignments using the "grow-diag-final-and" heuristic (Koehn et al., 2003). The significance test on word alignment performance was performed by the sign test with a 5% significance level. "+" in Table 2 indicates that the comparisons are significant over corresponding baselines, $IBM4$ and $FFNN_s(R/I)$.

In Table 2, RNN_{u+c}, which includes all our proposals, i.e., the RNN-based model, the unsupervised learning, and the agreement constraint, achieves the best performance for both $BTEC$ and $Hansards$. The differences from the baselines are statistically significant.

Table 2 shows that $RNN_s(R/I)$ outperforms $FFNN_s(R/I)$, which is statistically significant in $BTEC$. These results demonstrate that capturing the long alignment history in the RNN-based model improves the alignment performance. We discuss the difference of the RNN-based model's effectiveness between language pairs in Section 6.1. Table 2 also shows that $RNN_{s+c}(R/I)$ and RNN_{u+c} achieve significantly better performance than $RNN_s(R/I)$ and RNN_u in both tasks, respectively. This indicates that the proposed agreement constraint is effective in training better models in both the supervised and unsupervised approaches.

In $BTEC$, RNN_u and RNN_{u+c} significantly outperform $RNN_s(I)$ and $RNN_{s+c}(I)$, respectively. The performance of these models is comparable with $Hansards$. This indicates that our unsupervised learning benefits our models because the supervised models are adversely affected by errors in the automatically generated training data. This is especially true when the quality of training data, i.e., the performance of $IBM4$, is low.

5.4 Machine Translation Results

Table 3 shows the translation performance by the case sensitive BLEU4 metric[11] (Papineni et al., 2002). Table 3 presents the average BLEU of three different MERT runs. In $NTCIR$ and $FBIS$, each alignment model was trained from the ran-

[7] http://www.fit.vutbr.cz/~imikolov/rnnlm/

[8] http://chasen-legacy.sourceforge.jp/

[9] http://nlp.stanford.edu/software/segmenter.shtml

[10] Due to high computational cost, we did not use all the training data. Scaling up to larger datasets will be addressed in future work.

[11] We used mteval-v13a.pl as the evaluation tool (http://www.itl.nist.gov/iad/mig/tests/mt/2009/).

Alignment	$IWSLT$	$NTCIR$	$FBIS$	
			$NIST03$	$NIST04$
$IBM4_{all}$	46.47	27.91	25.90	28.34
$IBM4$		27.25	25.41	27.65
$FFNN_s$(I)	46.38	27.05	25.45	27.61
RNN_s(I)	46.43	27.24	25.47	27.56
RNN_{s+c}(I)	46.51	27.12	25.55	27.73
RNN_u	47.05*	27.79*	25.76*	27.91*
RNN_{u+c}	46.97*	27.76*	25.84*	28.20*

Table 3: Translation performance (BLEU4(%))

domly sampled 100 K data, and then a translation model was trained from all the training data that was word-aligned by the alignment model. In addition, for a detailed comparison, we evaluated the SMT system where the IBM Model 4 was trained from all the training data ($IBM4_{all}$). The significance test on translation performance was performed by the bootstrap method (Koehn, 2004) with a 5% significance level. "*" in Table 3 indicates that the comparisons are significant over both baselines, i.e., $IBM4$ and $FFNN_s(I)$.

Table 3 also shows that better word alignment does not always result in better translation, which has been discussed previously (Yang et al., 2013). However, RNN_u and RNN_{u+c} outperform $FFNN_s(I)$ and $IBM4$ in all tasks. These results indicate that our proposals contribute to improving translation performance[12]. In addition, Table 3 shows that these proposed models are comparable to $IBM4_{all}$ in $NTCIR$ and $FBIS$ even though the proposed models are trained from only a small part of the training data.

6 Discussion

6.1 Effectiveness of RNN-based Alignment Model

Figure 3 shows word alignment examples from $FFNN_s$ and RNN_s, where solid squares indicate the gold standard alignments. Figure 3 (a) shows that RRN_s adequately identifies complicated alignments with long distances compared to $FFNN_s$ (e.g., jaggy alignments of "have you been learning" in Fig 3 (a)) because RNN_s captures alignment paths based on long alignment history, which can be viewed as phrase-level alignments, while $FFNN_s$ employs only the last alignment.

In French-English word alignment, the most

[12]We also confirmed the effectiveness of our models on the NIST05 and NTCIR-10 evaluation data.

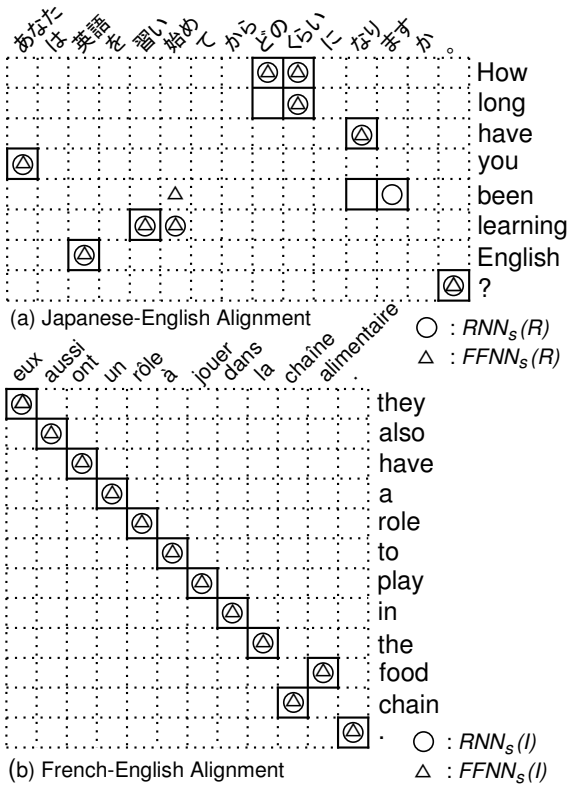

Figure 3: Word alignment examples

Alignment	40 K	9 K	1 K
$IBM4$	0.5467	0.4859	0.4128
RNN_{u+c}	0.6004	0.5562	0.4842
$RNN_{s+c}(R)$	-	0.8921	0.6063

Table 4: Word alignment performance on $BTEC$ with various sized training data

valuable clues are located locally because English and French have similar word orders and their alignment has more one-to-one mappings than Japanese-English word alignment (Figure 3). Figure 3 (b) shows that both RRN_s and $FFNN_s$ work for such simpler alignments. Therefore, the RNN-based model has less effect on French-English word alignment than Japanese-English word alignment, as indicated in Table 2.

6.2 Impact of Training Data Size

Table 4 shows the alignment performance on $BTEC$ with various training data sizes, i.e., training data for $IWSLT$ (40 K), training data for $BTEC$ (9 K), and the randomly sampled 1 K data from the $BTEC$ training data. Note that $RNN_{s+c}(R)$ cannot be trained from the 40 K data because the 40 K data does not have gold standard

Alignment	$BTEC$	$Hansards$
$FFNN_s$(I)	0.4770	0.9020
$FFNN_{s+c}$(I)	0.4854^+	0.9085^+
$FFNN_u$	0.5105^+	0.9026
$FFNN_{u+c}$	0.5313^+	0.9144^+
$FFNN_s$(R)	0.8224	-
$FFNN_{s+c}$(R)	0.8367^+	-

Table 5: Word alignment performance of various FFNN-based models (F1-measure)

word alignments.

Table 4 demonstrates that the proposed RNN-based model outperforms $IBM4$ trained from the unlabeled 40 K data by employing either the 1 K labeled data or the 9 K unlabeled data, which is less than 25% of the training data for $IBM4$. Consequently, the SMT system using RNN_{u+c} trained from a small part of training data can achieve comparable performance to that using $IBM4$ trained from all training data, which is shown in Table 3.

6.3 Effectiveness of Unsupervised Learning/Agreement Constraints

The proposed unsupervised learning and agreement constraints can be applied to any NN-based alignment model. Table 5 shows the alignment performance of the FFNN-based models trained by our supervised/unsupervised approaches (s/u) with and without our agreement constraints. In Table 5, "+c" denotes that the agreement constraint was used, and "+" indicates that the comparison with its corresponding baseline, i.e., $FFNN_s$(I/R), is significant in the sign test with a 5% significance level.

Table 5 shows that $FFNN_{s+c}$(R/I) and $FFNN_{u+c}$ achieve significantly better performance than $FFNN_s$(R/I) and $FFNN_u$, respectively, in both $BTEC$ and $Hansards$. In addition, $FFNN_u$ and $FFNN_{u+c}$ significantly outperform $FFNN_s$(I) and $FFNN_{s+c}$(I), respectively, in $BTEC$. The performance of these models is comparable in $Hansards$. These results indicate that the proposed unsupervised learning and agreement constraint benefit the FFNN-based model, similar to the RNN-based model.

7 Conclusion

We have proposed a word alignment model based on an RNN, which captures long alignment history through recurrent architectures. Furthermore, we proposed an unsupervised method for training our model using NCE and introduced an agreement constraint that encourages word embeddings to be consistent across alignment directions. Our experiments have shown that the proposed model outperforms the FFNN-based model (Yang et al., 2013) for word alignment and machine translation, and that the agreement constraint improves alignment performance.

In future, we plan to employ contexts composed of surrounding words (e.g., $c(f_j)$ or $c(e_{a_j})$ in the FFNN-based model) in our model, even though our model implicitly encodes such contexts in the alignment history. We also plan to enrich each hidden layer in our model with multiple layers following the success of Yang et al. (2013), in which multiple hidden layers improved the performance of the FFNN-based model. In addition, we would like to prove the effectiveness of the proposed method for other datasets.

Acknowledgments

We thank the anonymous reviewers for their helpful suggestions and valuable comments on the first version of this paper.

References

Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. 2013. Joint Language and Translation Modeling with Recurrent Neural Networks. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1044–1054.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155.

Phil Blunsom and Trevor Cohn. 2006. Discriminative Word Alignment with Conditional Random Fields. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 65–72.

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*, 19(2):263–311.

Ronan Collobert and Jason Weston. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In

Proceedings of the 25th International Conference on Machine Learning, pages 160–167.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537.

George E. Dahl, Dong Yu, Li Deng, and Alex Acero. 2012. Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Chris Dyer, Jonathan Clark, Alon Lavie, and Noah A. Smith. 2011. Unsupervised Word Alignment with Arbitrary Features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 409–419.

Cameron S. Fordyce. 2007. Overview of the IWSLT 2007 Evaluation Campaign. In *Proceedings of the 4th International Workshop on Spoken Languaeg Translation*, pages 1–12.

Kuzman Ganchev, João V. Graça, and Ben Taskar. 2008. Better Alignments = Better Translations? In *Proceedings of the 46th Annual Conference of the Association for Computational Linguistics: Human Language Technologies*, pages 986–993.

Chooi-Ling Goh, Taro Watanabe, Hirofumi Yamamoto, and Eiichiro Sumita. 2010. Constraining a Generative Word Alignment Model with Discriminative Output. *IEICE Transactions*, 93-D(7):1976–1983.

Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K. Tsou. 2011. Overview of the Patent Machine Translation Task at the NTCIR-9 Workshop. In *Proceedings of the 9th NTCIR Workshop*, pages 559–578.

João V. Graça, Kuzman Ganchev, and Ben Taskar. 2008. Expectation Maximization and Posterior Constraints. In *Advances in Neural Information Processing Systems 20*, pages 569–576.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-Contrastive Estimation: A New Estimation Principle for Unnormalized Statistical Models. In *Proceedings of the 13st International Conference on Artificial Intelligence and Statistics*, pages 297–304.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Human Language Technology Conference: North American Chapter of the Association for Computational Linguistics*, pages 48–54.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constrantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics on Interactive Poster and Demonstration Sessions*, pages 177–180.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395.

Hai-Son Le, Alexandre Allauzen, and François Yvon. 2012. Continuous Space Translation Models with Neural Networks. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–48.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by Agreement. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 104–111.

Evgeny Matusov, Richard Zens, and Hermann Ney. 2004. Symmetric Word Alignments for Statistical Machine Translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 219–225.

Rada Mihalcea and Ted Pedersen. 2003. An Evaluation Exercise for Word Alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 1–10.

Tomas Mikolov and Geoffrey Zweig. 2012. Context Dependent Recurrent Neural Network Language Model. In *Proceedings of the 4th IEEE Workshop on Spoken Language Technology*, pages 234–239.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent Neural Network based Language Model. In *Proceedings of 11th Annual Conference of the International Speech Communication Association*, pages 1045–1048.

Andriy Mnih and Yee Whye Teh. 2012. A Fast and Simple Algorithm for Training Neural Probabilistic Language Models. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1751–1758.

Robert C. Moore. 2005. A Discriminative Framework for Bilingual Word Alignment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 81–88.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29:19–51.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1986. Learning Internal Representations by Error Propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing*, pages 318–362. MIT Press.

Noah A. Smith and Jason Eisner. 2005. Contrastive Estimation: Training Log-Linear Models on Unlabeled Data. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 354–362.

Andreas Stolcke. 2002. SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of International Conference on Spoken Language Processing*, pages 901–904.

Martin Sundermeyer, Ilya Oparin, Jean-Luc Gauvain, Ben Freiberg, Ralf Schlüter, and Hermann Ney. 2013. Comparison of Feedforward and Recurrent Neural Network Language Models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 8430–8434.

Toshiyuki Takezawa, Eiichiro Sumita, Fumiaki Sugaya, Hirofumi Yamamoto, and Seiichi Yamamoto. 2002. Toward a Broad-coverage Bilingual Corpus for Speech Translation of Travel Conversations in the Real World. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 147–152.

Ben Taskar, Simon Lacoste-Julien, and Dan Klein. 2005. A Discriminative Matching Approach to Word Alignment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 73–80.

Ashish Vaswani, Liang Huang, and David Chiang. 2012. Smaller Alignment Models for Better Translations: Unsupervised Word Alignment with the l_0-norm. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 311–319.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with Large-Scale Neural Language Models Improves Translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1387–1392.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based Word Alignment in Statistical Translation. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 836–841.

Nan Yang, Shujie Liu, Mu Li, Ming Zhou, and Nenghai Yu. 2013. Word Alignment Modeling with Context Dependent Deep Neural Network. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 166–175.