Alignment by Bilingual Generation and Monolingual Derivation

Toshiaki Nakazawa *Sadao Kurohashi*
Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku
Kyoto, 606-8501, Japan
nakazawa@nlp.ist.i.kyoto-u.ac.jp, kuro@i.kyoto-u.ac.jp

ABSTRACT

One of the main issues in a word alignment task is the difficulty of handling function words that do not have direct translations which we call unique function words. They are often aligned to some words in the other language incorrectly. This is prominent in language pairs with very different sentence structures. In this paper, we propose a novel approach for handling unique function words. The proposed model *monolingually derives* unique function words from bilingually generated treelet pairs. The monolingual derivation prevents incorrect alignments for unique function words. The derivation probabilities are estimated from a large monolingual corpus, which is much easier to acquire than a parallel corpus. Also, the proposed alignment model uses semantic-head dependency trees where dependency relations between words become similar in each language. Experimental results on an English-Japanese corpus show that the proposed model achieves better alignment and translation quality compared with the baseline models.

TITLE AND ABSTRACT IN JAPANESE

二言語の生成と単言語の派生によるアライメント

単語アライメントタスクにおける主な問題の一つは、機能語の中でも相手言語に対応する語が存在しない機能語の扱いの困難さである。我々はこのような語を孤立機能語と呼ぶ。孤立機能語は、相手言語の何らかの単語に不適切に対応付けられることが多く、これは特に文構造が大きく異なる言語対において顕著である。本論文では、孤立機能語を扱うための新しい手法を提案する。提案モデルは、二言語で生成された部分木ペアから、孤立機能語をそれぞれ単言語で派生することにより、孤立機能語が誤って対応付けられることを防ぐ。派生確率は、対訳コーパスに比べて入手が容易である大規模単言語コーパスから推定する。また提案モデルは、単語同士の依存関係が各言語で近くなるように、意味主辞依存構造木を用いる。英日コーパスでの実験結果から、提案モデルはベースラインモデルと比べてより良いアライメントおよび翻訳精度を実現した。

KEYWORDS: monolingual derivation, semantic-head dependency tree, treelet alignment.

KEYWORDS IN JAPANESE: 単言語の派生, 意味主辞依存構造木, 木構造アライメント.

Proceedings of COLING 2012: Technical Papers, pages 1963–1978,
COLING 2012, Mumbai, December 2012.

1 Introduction

Alignment accuracy is crucial for providing high quality corpus-based machine translation systems because translation knowledge is acquired from an aligned training corpus. For similar language pairs, alignment accuracy is high. Less than 10% alignment error rate (AER) for French-English has been achieved by the conventional word alignment tool GIZA++, an implementation of the alignment models called the IBM models (Brown et al., 1993), with some heuristic symmetrization rules. However, for distant language pairs such as English-Japanese, the conventional alignment method is quite inadequate (achieving an AER of about 20%).

There are two main issues in a word alignment task for distant language pairs: one is the word order difference, while the other relates to function words. The word order issue has to some extent been solved by using word dependency trees in the alignment model (Nakazawa and Kurohashi, 2011). Most of the remaining alignment errors are related to function words such as English articles and Japanese case markers (Wu et al., 2011) because they do not have counterparts in the other language. As an example, most of the errors in Figure 1 are related to function words: "has" and "は (*topic-marker*)" in example (A), and "although", "は (*topic-marker*)", "を (*ACC*)" and "が (*but*)" in example (B).

Several previous works focused on alignment errors of function words. Isozaki et al. (2010) inserted pseudo nodes in English sentences for Japanese function words. Wu et al. (2011) removed the alignment of some function words to effectively acquire translation rules using the underlying word alignment by GIZA++. Nevertheless, these methods for dealing with function words are ad-hoc and based on hand-crafted rules.

In this paper, we propose a novel approach for handling function words. If there is a direct translation for a function word, these words should be aligned with each other. For function words that do not have any counterparts, the conventional model is supposed to align them to NULL, but it does not always work well. They are often aligned to some words incorrectly. In contrast with the conventional model, our model *derives* such function words from content words in their own language. The derivation probabilities used in our proposed model are estimated from a large monolingual corpus for each language. Thus, we do not require a large parallel corpus. With this derivation model, we can reduce alignment errors for function words, which leads to a better translation resources such as a phrase table, which is acquired from a word-aligned parallel corpus. In the remainder of this paper, we use English-Japanese language pairs for explanation. However, it should be noted that the proposed model is completely language independent.

2 Semantic-head dependency tree

The proposed model utilizes word dependency trees on both the source and target sides. Dependency trees are effective for language pairs with very different word orders, such as English-Japanese, to achieve high quality alignment by absorbing the difference (Nakazawa and Kurohashi, 2011). There are two types of word dependency trees: syntactic-head and semantic-head. Our model adopts the latter. This section discusses the difference between syntactic-head and semantic-head, and the reason why we choose the semantic-head dependency tree.

The syntactic-head dependency tree has two main drawbacks. One is that distances between content words are excessively large for agglutinative languages such as Japanese. The other is that dependency relations differ because of the difference in head word definitions in each

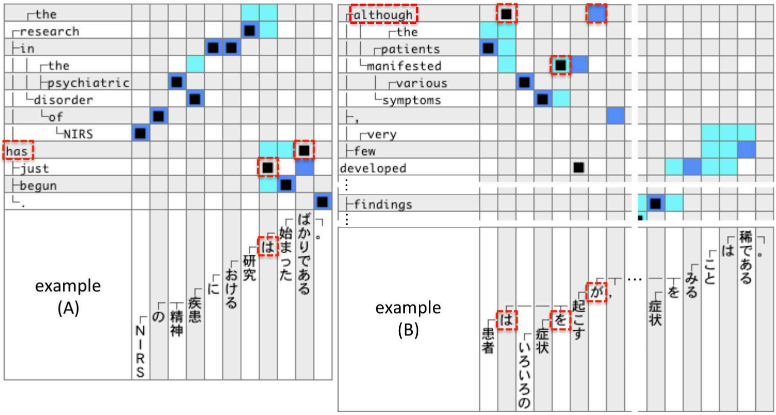

Figure 1: Alignment results of Nakazawa and Kurohashi (2011). Black boxes depict the system output, while dark blue (Sure) and light blue (Possible) cells denote gold-standard alignments. Cells demarcated by dotted lines are alignment errors related to function words.

language. On the other hand, in semantic-head dependency trees, function words giving additional information to content words are placed as children of the content words, thus it preserves the dependency relations between words over languages. In the semantic-head dependency tree (on the right of Figure 2), "medical treatment ⟷ 治療", "may ⟷ かも しれない", "not ⟷ ない" and "weight ⟷ 体重" are all children of "change ⟷ 変化", while the relations are not preserved in the syntactic-head dependency tree (on the left of Figure 2). Because of these advantages, our model uses semantic-head dependency trees.

In this paper, English sentences are first parsed by nlparser (Charniak and Johnson, 2005) which outputs phrase structures that are then converted into word dependency trees by defining the head word for phrases. The conversion rules follow Collins' head percolation table (Collins, 1999) with some modifications for acquiring semantic-head dependency trees. The following head-specifying rules are examples in which the syntactic head (underlined) and the semantic head (double underlined) is different.

- VP → $\underline{\text{MD}}$ $\underline{\underline{\text{VB}}}$ (ex. "may change" in Figure 2)
- VP → $\underline{\text{VBZ}}$ $\underline{\underline{\text{JJ}}}$ (ex. "is large")

Japanese sentences are usually parsed based on a unit called a *basic phrase*, which consists of one content word followed by zero or more function words. In syntactic-head word dependency trees, on the left of Figure 2, the head word is the last function word (or the content word if there is no function word in a basic phrase), and other words depend on their following words (Hajič et al., 2009). In semantic-head dependency trees, while function words showing a relationship between content words such as case markers are placed as parents of content words, other function words are placed as children. We obtain semantic-head dependency trees by modifying the rule file of the Japanese dependency analyzer KNP (Kawahara and Kurohashi, 2006b).

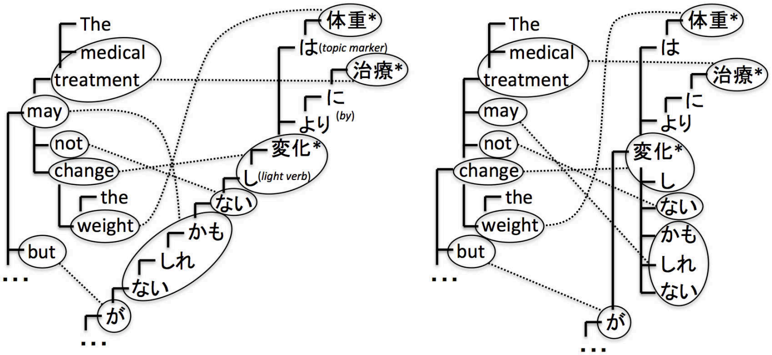

Figure 2: Examples of syntactic-head (left) and semantic-head (right) dependency trees. The root of the tree is at the extreme left and words are placed from top to bottom. Japanese content words are indicated by "*".

Several related studies use the semantic-head dependency trees in machine translation (Hong et al., 2009; Xu et al., 2009). They use English-side semantic-head dependency trees for pre-reordering of English sentences in order to make the word order closer to that of Subject-Object-Verb language sentences. The closer the word order is, the easier it is to train the model. However, certain hand-crafted rules are needed for reordering, and pseudo words are used to take care of function words. Compared with these studies, our proposed model uses semantic-head dependency trees on both sides and does not reorder sentences, and therefore it does not require any hand-crafted rules or pseudo words.

3 Handling function words

Although each language has a closed set of function words, the variety of function words differs between languages. For example, in Figure 2, some function words have direct translations in the other language ("may ⟷ かも しれ ない", "not ⟷ ない" and "but ⟷ が"), while others do not ("the", "は (*topic marker*)" and "に より (*by*)"). The first case is less problematic, and any alignment model can correctly detect the link. The second case represents the issue addressed in this paper. We call function words that do not have direct translations *unique function words*.

One solution, adopted by almost all the existing alignment models, is to align unique function words to NULL. However, it is difficult to judge whether a unique function word has to be NULL-aligned or not, and often causes alignment errors as shown in Figure 1. Also, the NULL-aligned words behave as gaps between aligned words, making it harder to capture relations between aligned words. In Figure 2, "に より (*by*)" is a gap between "治療 (*medical treatment*)" and "変化 し (*change*)" whereas counterparts have a direct parent-child relation.

Another solution is to enhance the alignment model so as to handle a larger unit than a word, and include the unique function words in neighboring alignments. For example, "weight ⟷ 体重" can include the Japanese "は (*topic marker*)" and become "weight ⟷ 体重 は". However, this solution can lead to a less appropriate parameter estimation for an alignment model because "weight ⟷ 体重" and "weight ⟷ 体重 は" are treated as different alignment patterns while the two patterns have essentially no difference.

The novel solution to this problem proposed in this paper is to *derive* unique function words from neighboring words monolingually. In the monolingual derivation model, "は (*topic marker*)" can be derived from "体重 (*weight*)" without changing the original treelet pair "weight ⟷ 体重", and therefore the model achieves good estimation of parameters. Note that it is also possible to derive "は (*topic marker*)" from "変化 し (*change*)" because they are contiguous in the tree structure. Another advantage of the monolingual derivation model is that it can reduce the gaps between alignments, and preserve the dependency relations between treelets over languages. For example, the model can derive "に より (*by*)" from "変化 し (*change*)" or " 治療 (*medical treatment*)" and let the dependency relation between "変化 し (*change*)" and "治療 (*medical treatment*)" be direct parent-child like their English counterparts. This is effective for estimating the dependency relation probability described in Section 4.3.

4 Model overview

The proposed model is an extension of that proposed by Nakazawa and Kurohashi (2011). This earlier model was overcoming the long-distance reordering issue by incorporating dependency trees. However, it was suffering from alignment errors for function words, which our new model solves by incorporating a *monolingual derivation model*. First we describe the generative story for the joint alignment model in the same manner as in previous work (Marcu and Wong, 2002; DeNero et al., 2008; Nakazawa and Kurohashi, 2011).

1. Generate ℓ concepts from which bilingual treelet pairs are generated independently.
2. For each treelet pair, derive zero or more treelets monolingually from each treelet in the treelet pair.
3. Combine the treelets in each language so as to create parallel sentences.

The number of concepts ℓ (> 0) is parameterized using a geometric distribution

$$P(\ell) = p_\$ \cdot (1 - p_\$)^{\ell - 1} \tag{1}$$

where $p_\$$ is a constant. Each concept generates a bilingual treelet pair from an unknown distribution θ_T. We call the treelet pair the *core alignment* denoting it as $\langle e_C, f_C \rangle$. Either one of the treelets in a treelet pair can be NULL, which represents an unaligned treelet. Unaligned treelets must be composed of exactly one word (NULL-alignment restriction).

Each treelet e_C and f_C derives sets of treelets $\{d_{e_C}\}$ and $\{d_{f_C}\}$ monolingually which basically consist of unique function words. The numbers of monolingual derivations $|\{d_{e_C}\}|$ and $|\{d_{f_C}\}|$ (≥ 0) are parameterized using a geometric distribution

$$P(|\{d_{e_C}\}|) = p_d \cdot (1 - p_d)^{|\{d_{e_C}\}|}, \quad P(|\{d_{f_C}\}|) = p_d \cdot (1 - p_d)^{|\{d_{f_C}\}|} \tag{2}$$

where p_d is a constant. Each derivation is drawn from a known multinomial distribution ϕ_{e_C} and ϕ_{f_C}, as explained in Section 4.2. We use the notation e to represent the combination of e_C and $\{d_{e_C}\}$, and f for f_C and $\{d_{f_C}\}$. Thus $\langle e, f \rangle$ contains $\langle e_C, f_C \rangle$, $\{d_{e_C}\}$ and $\{d_{f_C}\}$.

Finally, the treelet pairs are combined in each language. We denote the relation of treelets on the e-side as $D_E = \{(j \to k)\}$, where $(j \to k)$ denotes that treelet e_j depends on treelet e_k, and on the f-side as D_F. D refers to D_E and D_F as a whole. With these notations, the joint

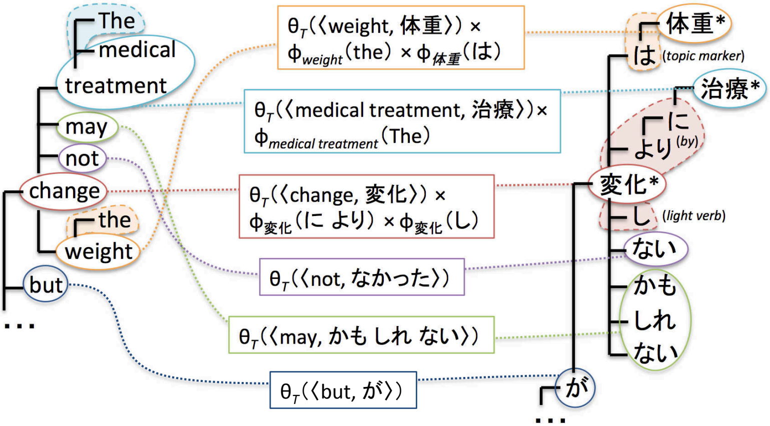

Figure 3: Example showing the calculation of the bilingual generation probability and monolingual derivation probability.

probability for an aligned sentence pair is defined as:

$$P(\ell,\{\langle e,f\rangle\},D)=P(\ell)\cdot P(D|\{\langle e,f\rangle\})\cdot\prod_{\langle e,f\rangle}\left(\theta_T(\langle e_C,f_C\rangle)\cdot P(|\{d_{e_C}\}|)\cdot\prod_{d_{e_C}}\phi_{e_C}(d_{e_C})\cdot P(|\{d_{f_C}\}|)\cdot\prod_{d_{f_C}}\phi_f(d_{f_C})\right). \tag{3}$$

In Figure 3, we show an example of the calculation of bilingual generation probability and monolingual derivation probability for each treelet pair (ignoring $P(|\{d_{e_C}\}|)$ and $P(|\{d_{f_C}\}|)$ for ease of understanding). For the core alignment ⟨not, ない⟩, there is no derivation, and only bilingual generation probability $\theta_T(\langle not, なかった\rangle)$ is used. For ⟨change, 変化⟩, there are two derivations from "変化 (*change*)"; "に より (*by*)" and "し (*light verb*)". Therefore, we need to calculate two monolingual derivation probabilities in addition to the bilingual generation probability.

The remainder of this section gives the details of the bilingual generation probability θ_T, monolingual derivation probability ϕ and dependency relation probability $P(D)$.

4.1 Bilingual generation probability

When generating bilingual treelets, we first need to decide whether to generate an unaligned treelet (with probability p_N) or an aligned treelet pair (with probability $1-p_N$). Aligned treelet pairs are generated from an unknown probability distribution θ_A, which obeys the Dirichlet process (DP):

$$\theta_A(\langle e_C,f_C\rangle) \quad \sim \quad DP(M_A,\alpha_A), \tag{4}$$

where M_A is the base distribution and α_A is a concentration parameter. The base distribution is defined as:

$$M_A(\langle e_C, f_C\rangle) = [P_e(e_C)P_{WA}(f_C|e_C) \cdot P_f(f_C)P_{WA}(e_C|f_C)]^{\frac{1}{2}}$$
$$P_e(e_C) = p_t \cdot (1-p_t)^{|e|-1} \cdot \left(\frac{1}{n_e}\right)^{|e_C|} \quad P_f(f_C) = p_t \cdot (1-p_t)^{|f|-1} \cdot \left(\frac{1}{n_f}\right)^{|f_C|}, \tag{5}$$

where P_{WA} is the translation probability computed by IBM model1 (Brown et al., 1993), and n_e and n_f are the numbers of word types in each language. θ_A does not give a weight to an unaligned treelet.

Unaligned treelets are generated from another unknown probability distribution θ_N:

$$\begin{array}{rcl} \theta_N(\langle e_C, f_C\rangle) & \sim & DP(M_N, \alpha_N) \\ M_N(\langle e_C, f_C\rangle) & = & \begin{cases} P_{WA}(e_C|\text{NULL}) & \text{if } f_C = \text{NULL} \\ P_{WA}(f_C|\text{NULL}) & \text{if } e_C = \text{NULL} \end{cases}. \end{array} \tag{6}$$

θ_N does not give a weight to an aligned treelet pair. Note that an unaligned treelet is always composed of only one word in our model. Finally, θ_T can be decomposed as:

$$\theta_T(\langle e_C, f_C\rangle) = p_N \theta_N(\langle e_C, f_C\rangle) + (1-p_N)\theta_A(\langle e_C, f_C\rangle). \tag{7}$$

The earlier study (Nakazawa and Kurohashi, 2011) only considered treelets as alignment units. However, this is inadequate for semantic-head dependency trees, since a set of sibling function words is often considered as an alignment unit. In Figure 2, for example, sibling "かも しれ ない (*may*)" in Japanese should be aligned to the English "may". Therefore, our model allows siblings to be a core alignment unit when the siblings are contiguous in the word sequence. We suppose the term "treelet" includes siblings in this paper.

4.2 Monolingual derivation probability

We only explain the e-side derivations in this section, since the f-side is the same. We calculate the monolingual derivation probability using a large monolingual corpus. Derivations d_{e_C} are conditioned on the treelet e_C from which they were derived:

$$\phi_e(d_{e_C}) = p(d_{e_C}|e_C). \tag{8}$$

For example, $\phi_{medical\ treatment}(\text{the}) = p(\text{the}|\text{medical treatment})$. However, using a treelet as a condition is vulnerable to the data sparseness problem. We use an *anchor word* in e_C as the condition instead of e_C. The derivation is connected to the anchor word in the word dependency tree:

$$p(d_{e_C}|e_C) \approx p(d_{e_C}|A(e_C, d_{e_C})). \tag{9}$$

The function $A(e_C, d_{e_C})$ returns the anchor word in e_C for d_{e_C}. For example, $A(\text{medical treatment}, \text{the})$ is "treatment".

$p(d_{e_C}|A(e_C, d_{e_C}))$ are calculated as follows:

$$p(d_{e_C}|A(e_C, d_{e_C})) = \frac{Count(d_{e_C}, A(e_C, d_{e_C}))}{\sum_d Count(d, A(e_C, d_{e_C}))}. \tag{10}$$

Anchor	Derivations
the	P:treatment, P:treatment change
medical	P:treatment, P:treatment change
treatment	L:the, L:medical, P:change
may	P:change
not	P:change
change	L:treatment, L:the treatment, L:medical treatment, L:the medical treatment, L:may, L:not, R:weight, R: the weight
the	P:weight, P: change weight
weight	P:change, L: the

Anchor	Derivations
体重	P:は, P:は 変化
は	L:体重, P:変化
治療	P:に, P:に より, P:に より 変化
に	L:治療, P:より, P:より 変化
より	L:に, L:治療 に, P:変化
変化	L:は, L:体重 は, L:より, L:に より, L:治療 に より, R:し, R:ない, R:かも, R:しれ, R:ない
し	P:変化
ない	P:変化

Table 1: The English derivations (left) and Japanese derivations (right) acquired from the sentences in Figure 3. 'P', 'L' and 'R' denote, respectively, Parent, pre-child (dependent from the Left), and post-child (dependent from the Right).

$Count(d_{e_C}, A(e_C, d_{e_C}))$ denotes the frequency with which d_{e_C} is connected to $A(e_C, d_{e_C})$ in the monolingual corpus. Taking each sentence in Figure 3 as an example sentence in the monolingual corpus, we can enumerate the derivations shown in Table 1 from the sentences. A derivation must be contiguous as a tree, and we do not consider sibling derivations. We distinguish three types of derivations: parent, pre-child (dependent from the left) and post-child (dependent from the right).

This lexicalized derivation is excessively specific. For example, the highest probability derivations from "Ph.D." acquired from the English Web corpus (Kawahara and Kurohashi, 2006a) are "a", "student", "thesis" in order. Consequently, using only lexicalized derivation can cause many derivation errors. We consider not only the lexicalized derivation probability, but also another probability using part-of-speech (POS) is used as the condition. Using the notation $A_{pos}(e_C, d_{e_C})$ for the POS of the anchor word, the monolingual derivation probability is defined as:

$$p(d_{e_C}|e_C) = [p(d_{e_C}|A(e_C, d_{e_C})) \cdot p(d_{e_C}|A_{pos}(e_C, d_{e_C}))]^{\frac{1}{2}}. \tag{11}$$

$p(d_{e_C}|A_{pos}(e_C, d_{e_C}))$ is also acquired from the large monolingual corpus in the same manner as $p(d_{e_C}|A(e_C, d_{e_C}))$. We take the geometric mean of the two probabilities because this eliminates noisy derivations of lexicalized probabilities while keeping the derivation preferences for each word.

Note that we do not need to discriminate between content words and function words in the enumeration of derivations and the calculation of derivation probabilities. Generally, the neighboring words of function words are content words. The vocabulary size of content words is much larger than that of function words. Therefore, the number of derivation patterns from function words is quite large, causing their probabilities to be very small. The probabilities naturally prefer deriving function words from content words than deriving content words from function words.

4.3 Dependency Relation Probability

Our model considers dependency relations between treelets and assigns a weight to each relation following the previous work (Nakazawa and Kurohashi, 2011). Here, each treelet includes both core and derivation treelets, and treelets in a treelet pair have the same index, for example, the counterpart of e_j is f_j.

The dependency relations are considered on each e and f side in the same manner, thus we only explain the e-side. First, we find the nearest aligned parent treelet, which we call relational parent, for each treelet in e-side. The relational parent is searched by ascending the dependency tree to the root node until an aligned treelet is found. The number of unaligned treelets on the path to relational parent from e_j is denoted as $N(e_j)$. $N(e_j) = 0$ if the relational parent is the direct parent of e_j. We consider an imaginary root as the relational parent for the root treelet of a sentence.

Suppose the relational parent treelet of e_j is e_k. Then, we consider where their counterparts, f_j and f_k respectively, are on the dependency tree of the other side. We can assume that f_j tends to depend on f_k because the dependencies between concepts hold across languages. The dependency relation probability reflects this tendency. We define the function $rel(e_j, e_k)$ which returns a dependency relation between the counterparts of the two arguments, in other words, dependency relation between f_j and f_k. We express a dependency relation as the shortest path from f_j to f_k. For simplicity, we indicate the path with a pair of non-negative integers, where the first is the number of steps going up (Up) the dependency tree and the other is the number going down ($Down$). For example, in Figure 3, traveling from “medical treatment” to “weight” requires 1 step going up (to reach “change”) and 1 step going down, so the dependency relation is $(Up, Down) = (1, 1)$.

Finally, we assign the dependency relation probability to a triplet of non-negative integers $R_f = (N, Up, Down)$. The dependency relation probabilities for the e-side are drawn from an unknown probability distribution θ_{ef} and for the f-side from θ_{fe}, with both obeying the DP:

$$\begin{array}{rclrcl} \theta_{ef}(R_e) & \sim & DP(M_{ef}, \alpha_{ef}) & M_{ef}(R_e) & = & p_{ef} \cdot (1 - p_{ef})^{N+Up+Down-1} \\ \theta_{fe}(R_f) & \sim & DP(M_{fe}, \alpha_{fe}) & M_{fe}(R_f) & = & p_{fe} \cdot (1 - p_{fe})^{N+Up+Down-1}. \end{array} \quad (12)$$

Using the notations and definitions above, the dependency tree-based reordering model $P(D|\{\langle e, f\rangle\})$ is decomposed as:

$$P(D|\{\langle e, f\rangle\}) = \prod_{\langle e,f\rangle} \theta_{ef}(R_e) \cdot \theta_{fe}(R_f). \quad (13)$$

5 Model training

We train the model by means of a collapsed Gibbs sampling, which has been used in some recent NLP works (Nakazawa and Kurohashi, 2011; DeNero et al., 2008). In a Gibbs sampling, we first need to initialize the states of the training data, such as the boundaries between treelets and their alignments, and also initialize the latent variables according to the initial states of the data. Starting with the initial state, we generate many samples sequentially from the last state by changing a small local point. Normalizing the counts in the samples yields the parameter estimations.

5.1 Initialization

We initialize the states of the training data by heuristically merging bi-directional alignment results of the standard word alignment tool GIZA++. Many machine translation studies use heuristics to combine the two alignment results, one of which is called grow-diag-final-and (Koehn et al., 2007). Our heuristic is similar to this, but the difference is that we combine the two results based on dependency trees, and not on word sequences. The initialization is carried out by the following steps:

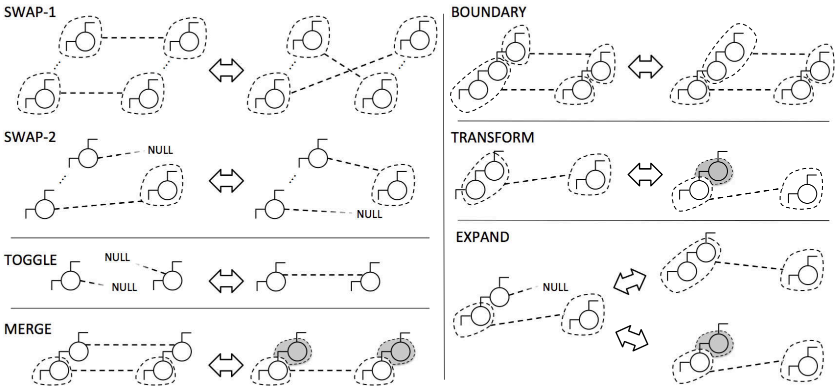

Figure 4: Illustration of the sampling operators. A solid circle represents a single word, while a treelet is depicted surrounded by a broken line. A gray treelet represents a derivation. A link directly connected to a word denotes that the treelet must consist of exactly one word, whereas other treelets can consist of one or more words including derivations.

1. Take the intersection of the two results.
2. In the union of the two results, accept alignment points connected to at least one accepted point in terms of the dependency tree (corresponds to grow-diag).
3. In the union of the two results, accept alignment points between two unaligned words (corresponds to final-and).

Initial boundaries of treelets and their alignments, and also the counts of treelet pairs and dependency relations are thus acquired. Note that there is no derivations after the initialization step.

5.2 Sampling operators

Our sampler repeatedly uses the six operators illustrated in Figure 4, to generate samples. Each application of an operator generates one new sample. We could, of course, use all the generated samples. However, since successive samples are almost the same, except for one local part, it is futile keeping all the samples. Thus, for each iteration, we keep only one sample, which is the final outcome after applying all the operators to all the possible points in all the sentence pairs in the training corpus.

SWAP
The SWAP operator exchanges the counterparts of two treelets, which may have derivations. There are two cases: [SWAP-1] both treelets are aligned, and [SWAP-2] one of the two treelets is unaligned and the other consists of exactly one word.

TOGGLE
The TOGGLE operator adds or removes an alignment. If f_j and e_k are both unaligned treelets, TOGGLE links the two treelets. Alternatively, if f_j and e_j are aligned, TOGGLE cuts the link

and makes each of the treelets unaligned. Because of the NULL-alignment restriction, f_j and e_j must consist of exactly one word.

MERGE
The MERGE operator combines a one-to-one alignment with the neighboring alignment as derivations from each treelet, or separates derivations from each treelet as an independent alignment.

BOUNDARY
The BOUNDARY operator moves the boundary between two treelets by one word. This operator does not change the type (core or derivation) of the boundary word.

TRANSFORM
The TRANSFORM operator changes the type of a word from core to derivation or vice versa.

EXPAND
The EXPAND operator expands or contracts an aligned treelet. If an unaligned treelet is next to an aligned one, EXPAND merges the unaligned and aligned treelets, either as a part of core treelet or derivation treelet. As the opposite direction, it excludes a marginal node from a treelet, and to make the excluded node unaligned.

6 Alignment experiments

We conducted alignment experiments on the English-Japanese corpus to show the effectiveness of the proposed model.

6.1 Settings

For the experiments, we used the JST[1] paper abstract corpus. This corpus was created by NICT[2] from JST's 2M English-Japanese paper abstract corpus using the method of Utiyama and Isahara (2007). This corpus consists of 996K parallel sentences: 24.7M words in English and 27.5M words in Japanese. Unfortunately, this corpus is not publicly available now, but they will become available in the near future.

As gold-standard data, 500 sentence pairs were annotated by hand using two types of annotations: sure (S) alignments and possible (P) alignments (Och and Ney, 2003). The unit of evaluation was the word. We used precision, recall, and alignment error rate (AER) as evaluation criteria. All the experiments were run on the original forms of words. The hyper parameters for our model used in the experiments are as follows: $p_\$ = 0.1$, $p_d = 0.9$, $p_N = 0.1$, $p_t = 0.8$, $\alpha_A = 100$, $\alpha_N = 100$, $\alpha_{fe} = 100$, $\alpha_{ef} = 100$, $p_{fe} = 0.5$, $p_{ef} = 0.5$. They are borrowed from the previous work (DeNero et al., 2008; Nakazawa and Kurohashi, 2011) and changed a little. The training time was about 1 day using 200 CPU cores. It is much slower than the word-sequence-based models because considering tree structures is computationally more complex.

The derivation probabilities were calculated from English and Japanese Web corpora each consisting of 550M sentences (Kawahara and Kurohashi, 2006a). We limited the maximum size of a derivation treelet to three words. We only consider top-20 frequent derivations for each word and POS.

[1] http://www.jst.go.jp/

[2] http://www.nict.go.jp/

English sentences were converted into phrase structures using Charniak's nlparser (Charniak and Johnson, 2005), and then they were transformed into dependency structures by rules defining head words for phrases (Collins, 1999). Japanese sentences were converted into dependency structures using the morphological analyzer JUMAN (Kurohashi et al., 1994) and the dependency analyzer KNP (Kawahara and Kurohashi, 2006b).

For comparison, we used GIZA++ (Och and Ney, 2003), which implements the well-known word-based statistical alignment model of the IBM Models. We conducted word alignment bidirectionally with the default parameters and merged them using the grow-diag-final-and heuristics (Koehn et al., 2003). We also tested the BerkeleyAligner[3] (DeNero and Klein, 2007) in the unsupervised training mode with default settings.

6.2 Experimental result and discussion

The experimental results are given in Table 2. "Syntactic-head" is the alignment accuracy of the baseline system by Nakazawa and Kurohashi (2011), while "Semantic-head w/o derivation" is the result of using the baseline model on semantic-head dependency trees. The results of incorporating the monolingual derivation are given in the bottom two rows, where "all" means that we evaluated all the alignments including derivations, while "core" means that we only evaluated the core alignments.

As mentioned in Section 1, the baseline model has already shown much better alignment accuracy than the conventional models, GIZA++ and BerkeleyAligner. There was a slight improvement using semantic-head dependency trees (0.36% absolute AER reduction).

The proposed model further improved the alignment accuracy. Compared with the baseline model, we achieved 0.6% and 1.34% improvement in absolute AER by evaluating all the alignments and only the core alignments respectively. The relative error reduction in AER is about 10% for the core alignment, which can be considered as a significant improvement. The reason of the further AER decrease when using only the core alignments is as follows: although the monolingual derivation can prevent from incorrect alignments for unique function words, it sometimes causes over derivations. This is discussed in detail later.

Figure 5 shows the alignment results by the proposed model for sentences in Figure 1. The proposed model reduced the alignment errors for unique function words by deriving them monolingually, and found correct alignments which the baseline system failed to find.

There are two main causes of alignment errors in the proposed model. One is the granularity of the derivation probability. We used the product of the two probabilities, lexicalized and POS-based, to take advantage of them, but this is insufficient. For example, in the sentence fragment "the possibility that ...", the proposed model failed to derive the unique word "that" from "possibility". In another fragment "the patient who ...", the proposed model failed to derive "who" from "patient". The reason for these failures is the low probability in the POS-based derivation. The possible solution for the granularity problem is to use word classes where each word class contains the words which have similar derivation distribution. For example, some abstract nouns such as "possibility" and "fact" tend to derive appositive "that", and person-category nouns tend to derive a relative "who".

The other cause of alignment errors is the over derivation typically created by the noise in a parallel sentence and parsing error. On the left of Figure 6, the fragment of the Japanese

[3] http://code.google.com/p/berkeleyaligner/

Alignment model	Pre.	Rec.	AER
GIZA++ & grow-diag-final-and	83.00	83.01	16.99
BerkeleyAligner	85.69	74.13	20.32
Syntactic-head (baseline)	88.59	83.78	13.66
Semantic-head w/o derivation	88.82	84.28	13.30
Semantic-head w/ derivation (all)	87.83	**85.81**	13.06
Semantic-head w/ derivation (core)	**90.51**	84.49	**12.32**

Table 2: Results of English-Japanese alignment experiments.

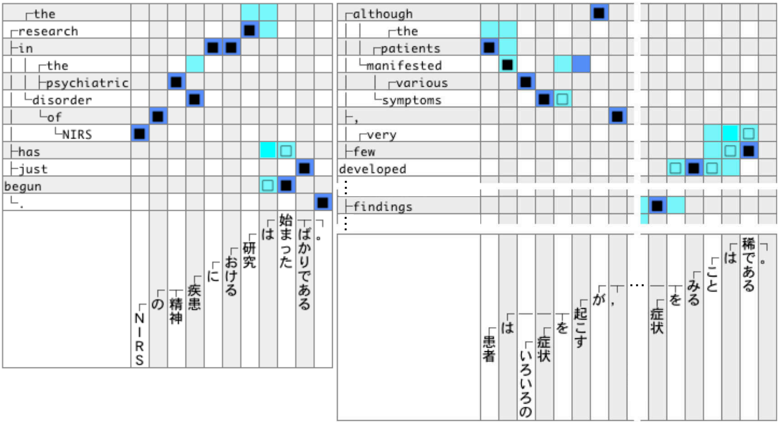

Figure 5: Alignment results by the proposed model. The solid and open squares depict core alignments and derivations, respectively.

sentence “音 列 の (*of tone sequence*)” has no corresponding part in the English sentence. “音列 (*tone sequence*)” is correctly aligned to NULL, but the function word “の (*of*)” is incorrectly derived from “法 (*method*)”. On the right of Figure 6, the Japanese “HDMI は” should not depend on “異なる (*different*)”, but on “使用 (*use*)”. Because of this parsing error, the topic marker “は” is incorrectly derived from “異なる (*different*)”. One possible short-term solution for the parsing problem is to use the n-best parsing results in the model. An alternative solution was proposed by Burkett et al. (2010), who described a joint parsing and alignment model that exchanges useful information between the parser and aligner.

7 Translation experiments

We conducted English-to-Japanese and Japanese-to-English translation experiments on the same corpus used in the alignment experiments. We translated 500 paper abstract sentences from the JST corpus. Note that these sentences were not included in the training corpus. We use Joshua[4], a Java-based opensource implementation of the hierarchical decoder, version 4.0 (Ganitkevitch et al., 2012) with default settings. It was tuned using another 500 development sentence pairs.

Table 3 shows the BLEU (Papineni et al., 2002) scores for the translations. The proposed

[4] http://joshua-decoder.org

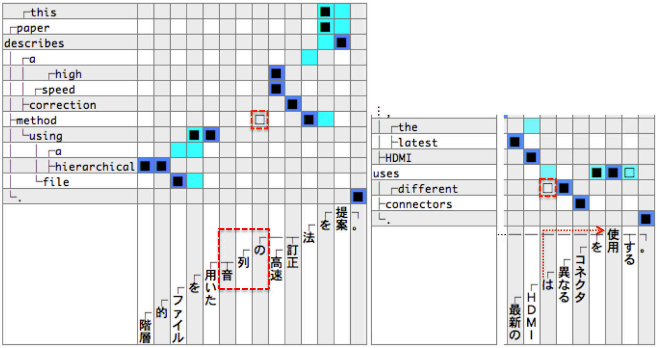

Figure 6: Alignment errors of the proposed model caused by a NULL part (left) and a parsing error (right).

Alignment model	En → Ja	Ja → En
GIZA++ & grow-diag-final-and	23.84	17.75
Syntactic-head (baseline)	24.16	17.83
Semantic-head w/o derivation	24.11	18.06
Semantic-head w/ derivation (all)	**24.55**†	**18.46**†‡
Semantic-head w/ derivation (core)	24.45	17.76

Table 3: BLEU scores for English-to-Japanese and Japanese-to-English translation experiments. † and ‡ marks indicate significant difference by bootstrap resampling (Koehn, 2004) from the decoder using GIZA++ & grow-diag-final-and alignment and baseline alignment respectively ($p < 0.05$).

model using all the alignments including derivations achieved the best translation quality. We believe this improvement is due to the reduction in function word alignment errors. The BLEU score decreased when only core alignments were used. This is because the exclusion of the derivations increased the ambiguity of the translation rules.

Conclusion and future work

In this paper, we proposed a novel approach for handling unique function words based on semantic-head dependency trees. The proposed model monolingually derives unique function words from bilingually generated treelet pairs. The derivation probabilities are acquired from a large monolingual corpus for each language. We showed that semantic-head dependency trees are more effective than syntactic-head dependency trees for high quality alignment, and that the treelet derivation model can reduce alignment errors for function words resulting in better translation quality.

To further validate the effectiveness of the proposed model, we need to apply our model to other language pairs, including the Korean language, which is also an agglutinative language. In addition, we need to resolve the issues discussed in Section 6.2.

Acknowledgments

This work was partially supported by Yahoo Japan Corporation.

References

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Association for Computational Linguistics*, 19(2):263–312.

Burkett, D., Blitzer, J., and Klein, D. (2010). Joint parsing and alignment with weakly synchronized grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–135, Los Angeles, California. Association for Computational Linguistics.

Charniak, E. and Johnson, M. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180.

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania.

DeNero, J., Bouchard-Côté, A., and Klein, D. (2008). Sampling alignment structure under a Bayesian translation model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 314–323, Honolulu, Hawaii. Association for Computational Linguistics.

DeNero, J. and Klein, D. (2007). Tailoring word alignments to syntactic machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 17–24, Prague, Czech Republic. Association for Computational Linguistics.

Ganitkevitch, J., Cao, Y., Weese, J., Post, M., and Callison-Burch, C. (2012). Joshua 4.0: Packing, pro, and paraphrases. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 283–291, Montréal, Canada. Association for Computational Linguistics.

Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., and Zhang, Y. (2009). The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado. Association for Computational Linguistics.

Hong, G., Lee, S.-W., and Rim, H.-C. (2009). Bridging morpho-syntactic gap between source and target sentences for english-korean statistical machine translation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 233–236, Suntec, Singapore. Association for Computational Linguistics.

Isozaki, H., Sudoh, K., Tsukada, H., and Duh, K. (2010). Head finalization: A simple reordering rule for sov languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 244–251, Uppsala, Sweden. Association for Computational Linguistics.

Kawahara, D. and Kurohashi, S. (2006a). Case frame compilation from the web using high-performance computing. In *the 5th International Conference on Language Resources and Evaluation (LREC2006)*.

Kawahara, D. and Kurohashi, S. (2006b). A fully-lexicalized probabilistic model for japanese syntactic and case structure analysis. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 176–183, New York City, USA. Association for Computational Linguistics.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In Lin, D. and Wu, D., editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

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. (2007). Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *HLT-NAACL 2003: Main Proceedings*, pages 127–133.

Kurohashi, S., Nakamura, T., Matsumoto, Y., and Nagao, M. (1994). Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of The International Workshop on Sharable Natural Language*, pages 22–28.

Marcu, D. and Wong, D. (2002). A phrase-based,joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 133–139. Association for Computational Linguistics.

Nakazawa, T. and Kurohashi, S. (2011). Bayesian subtree alignment model based on dependency trees. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP2011)*, pages 794–802, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Association for Computational Linguistics*, 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Utiyama, M. and Isahara, H. (2007). A japanese-english patent parallel corpus. In *MT summit XI*, pages 475–482.

Wu, X., Matsuzaki, T., and Tsujii, J. (2011). Effective use of function words for rule generalization in forest-based translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 22–31, Portland, Oregon, USA. Association for Computational Linguistics.

Xu, P., Kang, J., Ringgaard, M., and Och, F. (2009). Using a dependency parser to improve smt for subject-object-verb languages. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 245–253, Boulder, Colorado. Association for Computational Linguistics.