Dependency-based Automatic Enumeration of Semantically Equivalent Word Orders for Evaluating Japanese Translations

Hideki Isozaki, Natsume Kouchi
Okayama Prefectural University
111 Kuboki, Soja-shi, Okayama, 719-1197, Japan
isozaki@cse.oka-pu.ac.jp

Tsutomu Hirao
NTT Communication Science Laboratories
2-4, Hikaridai, Seika-cho, Sorakugun, Kyoto, 619-0237, Japan
hirao.tsutomu@lab.ntt.co.jp

Abstract

Scrambling is acceptable reordering of verb arguments in languages such as Japanese and German. In automatic evaluation of translation quality, BLEU is the de facto standard method, but BLEU has only very weak correlation with human judgements in case of Japanese-to-English/English-to-Japanese translations. Therefore, alternative methods, IMPACT and RIBES, were proposed and they have shown much stronger correlation than BLEU. Now, RIBES is widely used in recent papers on Japanese-related translations. RIBES compares word order of MT output with manually translated reference sentences but it does not regard scrambling at all. In this paper, we present a method to enumerate scrambled sentences from dependency trees of reference sentences. Our experiments based on NTCIR Patent MT data show that the method improves sentence-level correlation between RIBES and human-judged adequacy.

1 Introduction

Statistical Machine Translation has grown with an automatic evaluation method BLEU (Papineni et al., 2002). BLEU measures local word order by n-grams and does not care about global word order. In JE/EJ translations, this insensitivity degrades BLEU's correlation with human judgements.

Therefore, alternative automatic evaluation methods are proposed. Echizen-ya and Araki (2007) proposed IMPACT. Isozaki et al. (2010) presented the idea of RIBES. Hirao et al. (2011) named this method "RIBES" (Rank-based Intuitive Bilingual Evaluation Score). This version of RIBES was defined as follows:

$$\text{RIBES} = \text{NKT} \times P^{\alpha}$$

Table 1: Meta-evaluation of NTCIR-7 JE task data (Spearman's ρ, System-level correlation)

BLEU	METEOR	ROUGE-L	IMPACT	**RIBES**
0.515	0.490	0.903	0.826	**0.947**

where NKT (Normalized Kendall's τ) is defined by $(\tau + 1)/2$. This NKT is used for measuring word order similarity between a reference sentence and an MT output sentence. Thus, RIBES penalizes difference of global word order. P is precision of unigrams. RIBES is defined for each test sentence and averaged RIBES is used for evaluating the entire test corpus.

Table 1 is a table in an IWSLT-2012 invited talk (http://hltc.cs.ust.hk/iwslt/slides/Isozaki2012_slides.pdf). METEOR was proposed by Banerjee and Lavie (2005). ROUGE-L was proposed by Lin and Och (2004). According to this table, RIBES with $\alpha = 0.2$ has a very strong correlation (Spearman's ρ = 0.947) with human-judged adequacy. For each sentence, we use the average of adequacy scores of three judges. Here, we call this average "Adequacy". We focus on Adequacy because current SMT systems tend to output inadequate sentences. Note that only single reference translations are available for this task although use of multiple references is common for BLEU.

RIBES is publicly available from http://www.kecl.ntt.co.jp/icl/lirg/ribes/ and was used as a standard quality measure in recent NTCIR PatentMT tasks (Goto et al., 2011; Goto et al., 2013). Table 2 shows the result of meta-evaluation at NICTR-9/10 PatentMT. The table shows that RIBES is more reliable than BLEU and NIST.

Current RIBES has the following improvements.

- BLEU's Brevity Penalty (BP) was introduced

Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 287–292,
Baltimore, Maryland USA, June 26–27, 2014.

Table 2: Meta-evaluation at NTCIR-9/10 PatentMT (Spearman's ρ, Goto et al. 2011, 2013)

	BLEU	NIST	**RIBES**
NTCIR-9 JE	-0.042	-0.114	**0.632**
NTCIR-9 EJ	-0.029	-0.074	**0.716**
NTCIR-10 JE	0.31	0.36	**0.88**
NTCIR-10 EJ	0.36	0.22	**0.79**

in order to penalize too short sentences.

$$\text{RIBES} = \text{NKT} \times P^{\alpha} \times \text{BP}^{\beta}$$

where $\alpha = 0.25$ and $\beta = 0.10$. BLEU uses BP for the entire test corpus, but RIBES uses it for each sentence.

- The word alignment algorithm in the original RIBES used only bigrams for disambiguation when the same word appears twice or more in one sentence. This restriction is now removed, and longer n-grams are used to get a better alignment.

RIBES is widely used in recent Annual Meeings of the (Japanese) Association for NLP. International conference papers on Japanese-related translations also use RIBES. (Wu et al., 2012; Neubig et al., 2012; Goto et al., 2012; Hayashi et al., 2013). Dan et al. (2012) uses RIBES for Chinese-to-Japanese translation.

However, we have to take "*scrambling*" into account when we think of Japanese word order. Scrambling is also observed in other languages such as German. Current RIBES does not regard this fact.

2 Methodology

For instance, a Japanese sentence S1

jon **ga** sushi-ya **de** o-sushi **wo** tabe-ta .
(John ate sushi at a sushi restaurant.)

has the following acceptable word orders.

1. jon **ga** sushi-ya **de** o-sushi **wo** tabe-ta .
2. jon **ga** o-sushi **wo** sushi-ya **de** tabe-ta .
3. sushi-ya **de** jon **ga** o-sushi **wo** tabe-ta .
4. sushi-ya **de** o-sushi **wo** jon **ga** tabe-ta .
5. o-sushi **wo** jon **ga** sushi-ya **de** tabe-ta .
6. o-sushi **wo** sushi-ya **de** jon **ga** tabe-ta .

The boldface short words "**ga**", "**de**", and "**wo**", are *case markers* ("*Kaku joshi*" in Japanese).

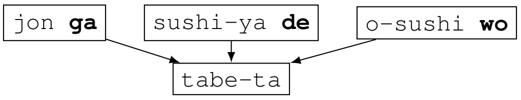


Figure 1: Dependency Tree of S1

- "**ga**" is a nominative case marker that means the noun phrase before it is the subject of a following verb/adjective.
- "**de**" is a locative case marker that means the noun phrase before it is the location of a following verb/adjective.
- "**wo**" is an accusative case marker that means the noun phrase before it is the direct object of a following verb.

The term "*scrambling*" stands for these acceptable permutations. These case markers explicitly show grammatical cases and reordering of them does not hurt interpretation of these sentences. Almost all other permutations of words are not acceptable ($*$).

$*$ jon **ga de** sushi-ya o-sushi tabe-ta **wo** .
$*$ jon **de** sushi-ya **ga** o-sushi **wo** tabe-ta .
$*$ jon tabe-ta **ga** o-sushi **wo** sushi-ya **de** .
$*$ sushi-ya **ga** jon tabe-ta **de** o-sushi **wo** .

Most readers unfamiliar with Japanese will not understand which word order is acceptable.

2.1 Scrambling as Post-Order Traversal of Depenedncy Trees

Here, we describe this "*scrambling*" from the viewpoint of Computer Science. Figure 1 shows S1's dependency tree. Each box indicates a "*bunsetsu*" or a grammatical chunk of words. Each arrow starts from a modifier (dependent) to its head.

The root of S1 is "tabe-ta" (ate). This verb has three modifiers:

- "jon **ga**" (John is its subject)
- "sushi-ya **de**" (A sushi restaurant is its location)
- "o-sushi **wo**" (Sushi is its object)

It is well known that Japanese is a typical head-final language. In order to generate a head-final word order from this dependency tree, we should output tree nodes in **post-order**. That is, we have to output all children of a node N before the node N itself.

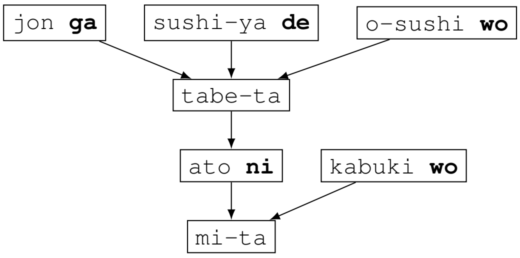


Figure 2: Dependency Tree of S2

All of the above acceptable word orders follows this post-order. Even in post-order traverse, precedence among children is not determined and this fact leads to different permutations of children. In the above example, the root "`tabe-ta`" has three children, and its permutation is 3! = 6.

2.2 Simple Case Marker Constraint

Figure 2 shows the dependency tree of a more complicated sentence S2:

> `jon` **`ga`** `sushi-ya` **`de`** `o-sushi` **`wo`** `tabe-ta ato` **`ni`** `kabuki` **`wo`** `mi-ta .`
> (John watched kabuki after eating sushi at a shushi restaurant)

Kabuki is a traditional Japanese drama performed in a theatre. In this case, the root "`mi-ta`" (watched) has two children: "`ato` **`ni`**" (after it) and "`kabuki` **`wo`**" (kabuki is its object).

- "**`ni`**" is a dative/locative case marker that means the noun phrase before it is an indirect object or a location/time of a following verb/adjective.

In this case, we obtain 3! × 2! = 12 permutations:

1. *S1P* `ato` **`ni`** `kabuki` **`wo`** `mi-ta .`
2. `kabuki` **`wo`** *S1P* `ato` **`ni`** `mi-ta .`

Here, *S1P* is any of the above 3! permutations of S1. If we use S1's 3 as *S1P* in S2's 1, we get

> `sushi-ya` **`de`** `jon` **`ga`** `o-sushi` **`wo`** `tabe-ta ato` **`ni`** `kabuki` **`wo`** `mi-ta .`

However, we cannot accept all of these permutations equally. For instance,

> `kabuki` **`wo`** `o-sushi` **`wo`** `sushi-ya` **`de`** `jon` **`ga`** `tabe-ta ato` **`ni`** `mi-ta .`

is comprehensible but strange. This strangeness comes from the two objective markers "`wo`" before the first verb "`tabe-ta`." Which did John eat, kabuki or sushi? Semantically, we cannot eat kabuki (drama), and we can understand this sentence. But syntactic ambiguity causes this strangeness. Without semantic knowledge about kabuki and sushi, we cannot disambiguate this case.

For readers/listeners, we should avoid such syntactically ambiguous sentences. Modifiers (here, "`kabuki` **`wo`**") of a verb (here, "`mi-ta`", watched) should not be placed before another verb (here, "`tabe-ta`", ate).

In Japanese, verbs and adjectives are used similarly. In general, adjectives are not modified by "**`wo`**" case markers. Therefore, we can place "**`wo`**" case markers before adjectives. In the following sentences, "`atarashii`" (new) is an adjective and placing "`inu` **`wo`**" (A dog is the direct object) before "`atarashii`" does not make the sentence ambiguous.

- *`atarashii`* `ie` **`ni`** `inu` **`wo`** `ture te itta .`

((Someone) took the dog to the new house.)

- `inu` **`wo`** *`atarashii`* `ie` **`ni`** `ture te itta .`

This idea leads to the following Simple Case Marker Constraint:

Definition 1 (Simple Case Marker Constraint) *If a reordered sentence has a case marker phrase of a verb that precedes another verb before the verb, the sentence is rejected. "**`wo`**" case markers can precede adjectives before the verb.*

This is a primitive heuristic constraint and there must be better ways to make it more flexible. If we use Nihongo Goi Taikei (Ikehara et al., 1997), we will be able to implement such a flexible constraint. For example, some verbs such as "`sai-ta`" (bloomed) are never modified by "**`wo`**" case marker phrases. Therefore, the following sentence is not ambiguous at all although the **`wo`** phrase precedes "`sai-ta`".

- `hana` **`ga`** `sai-ta ato` **`ni`** `sono ki` **`wo`** `mi-ta.`

((Someone) saw the tree after it bloomed.)

- `sono ki` **`wo`** `hana` **`ga`** `sai-ta ato` **`ni`** `mi-ta.`

2.3 Evaluation with scrambled sentences

As we mentioned before, RIBES measures global word order similarity between machine-translated sentences and reference sentences. It does not regard scrambling at all. When the target language allows scrambling just like Japanese, RIBES should consider scrambling.

Once we have a correct dependency tree of the reference sentence, we enumerate scrambled sentences by reordering children of each node. The number of the reordered sentences depend on the structure of the dependency tree.

Current RIBES code (RIBES-1.02.4) assumes that every sentence has a fixed number of references, but here the number of automatically generated reference sentences depends on the dependency structure of the original reference sentence. Therefore, we modified the code for variable numbers of reference sentences. RIBES-1.02.4 simply uses the maximum value of the scores for different reference sentences, and we followed it.

Here, we compare the following four methods.

- single: We use only single reference translations provided by the NTCIR organizers.

- postOrder: We generate all permutations of the given reference sentence generated by post-order traversals of its dependency tree. This can be achieved by the following two steps. First, we enumerate all permutations of child nodes at each node. Then, we combine these permutations. This is implemented by cartesian products of the permutation sets.

- caseMarkers: We reorder only "case marker (*kaku joshi*) phrases". Here, a "case marker phrase" is post-order traversal of a subtree rooted at a case marker *bunsetsu*. For instance, the root of the following sentence S3 has a non-case marker child "kaburi ," (wear) between case marker children, "jon **ga**" and "zubon **wo**" (Trousers are the object). Figure 3 shows its dependency tree.

  jon **ga** shiroi boushi **wo** kaburi ,
  kuroi zubon **wo** hai te iru.
  (John wears a white hat and wears black trousers.)

  This is implemented by removing non-case marker nodes from the set of child nodes to be reordered in the above "postOrder" method. For simplicity, we do not reorder other markers such as the topic marker "**wa**" here. This is future work.

- proposed: We reorder only *contiguous* case marker children of a node, and we accept sentences that satisfy the aforementioned Simple Case Marker Constraint. S3's root node has two case marker children, but they are not contiguous. Therefore, we do not reorder them. We expect that the constraint inhibit generation of incomprehensible or misleading sentences.

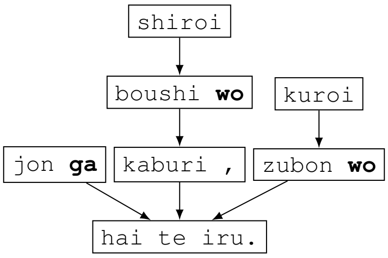


Figure 3: Dependency Tree of S3

Table 3: Distribution of the number of generated permutations

#permutations	1	2	4	6	8	12	16	24	>24
single	100	0	0	0	0	0	0	0	0
proposed	70	20	7	3	0	0	0	0	0
caseMarkers	64	23	4	6	2	2	0	2	0
postOrder	1	17	9	11	4	12	1	12	33

3 Results

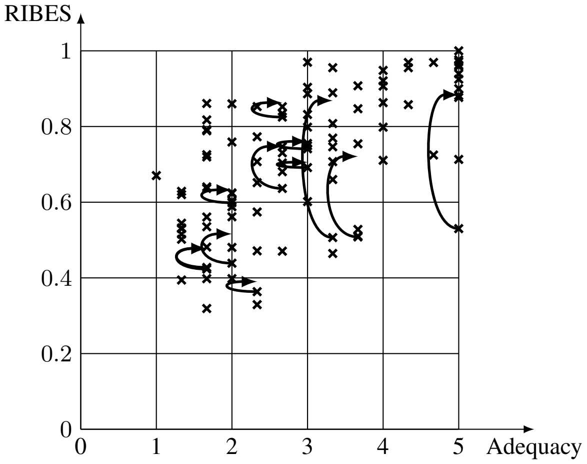
We applied the above four methods to the reference sentences of human-judged 100 sentences of NTCIR-7 Patent MT EJ task. (Fujii et al., 2008) We applied CaboCha (Kudo and Matsumoto, 2002) to the reference sentences, and manually corrected the dependency trees because Japanese dependency parsers are not satisfactory in terms of sentence accuracy (Tamura et al., 2007).

To support this manual correction, CaboCha's XML output was automatically converted to dependency tree pictures by using cabochatrees package for LaTeX. http://softcream.oka-pu.ac.jp/wp/wp-content/uploads/cabochatrees.pdf. Then, it is easy to find mistakes of the dependency trees. In addition, CaboCha's dependency accuracy is very high (89–90%) (Kudo and Matsumoto, 2002). Therefore, it took only one day to fix dependency trees of one hundred reference sentences.

Table 3 shows distribution of the number of word orders generated by the above methods. PostOrder sometimes generates tens of thousands of permutations.

Figure 4 shows a sentence-level scatter plot between Adequacy and RIBES for the baseline Moses system. Each × indicates a sentence.

Arrows indicate significant improvements of RIBES scores by the proposed method. For instance, the × mark at (5.0, 0.53) corresponds to an MT output:



Average of RIBES: 0.706 → 0.719
Pearson's r: 0.607 → 0.663
Spearman's ρ: 0.607 → 0.670

Figure 4: Scatter plot between Adequacy and RIBES for 100 human-judged sentences in the output of NTCIR-7's baseline Moses system and the effects of the proposed method

```
indekkusu kohna wo zu 25 ni shimesu .
```

which is a Japanese translation of "FIG.25 shows the index corner." The reference sentence for this sentence is

```
zu 25 ni indekkusu kohna wo shimeshi
te iru .
```

In this case, RIBES is 0.53, but all of the three judges evaluated this as 5 of 5-point scale. That is, RIBES disagrees with human judges. The proposed method reorders this reference sentence as follows:

```
indekkusu kohna wo zu 25 ni shimeshi
te iru .
```

This is very close to the above MT output and RIBES is 0.884 for this automatically reordered reference sentence. This shows that automatic reordering reduces the gap between single-reference RIBES and Adequacy.

Although RIBES strongly correlates with adequacy at the system level (Table 1), it has only mediocre correlation with adequacy at the sentence level: Spearman's ρ is 0.607 for the baseline Moses system. The "proposed" method improves it to 0.670.

We can draw similar scatter plots for each system. **Table 4** summarises such improvement of correlations. And this is the main result of this paper. The "**proposed**" method consistently improves sentence-level correlation between Adequacy and RIBES.

Table 4: Improvement of sentence-level correlation between Adequacy and RIBES for human-judged NTCIR-7 EJ systems (MAIN RESULT)

	Pearson's r	Spearman's ρ
	single → **proposed**	single → **proposed**
tsbmt	0.466 → 0.472	0.439 → 0.452
Moses	0.607 → 0.663	0.607 → 0.670
NTT	0.709 → 0.735	0.692 → 0.727
NICT-ATR	0.620 → 0.631	0.582 → 0.608
kuro	0.555 → 0.608	0.515 → 0.550

Table 5: Increase of averaged RIBES scores

	Adeq.	RIBES			
system		single	**proposed**	caseMarkers	postOrder
tsbmt	3.527	0.715	0.718$_8$	0.719	0.756$_9$
moses	2.897	0.706	0.719$_2$	0.722	0.781
NTT	2.740	0.671	0.683	0.686	0.756$_5$
NICT-ATR	2.587	0.655	0.664	0.670	0.749
kuro	2.420	0.629	0.638	0.647	0.752

Table 5 shows increase of averaged RIBES, but this increase is not always an improvement. We expected that "PostOrder" generates not only acceptable sentences but also incomprehensible or misleading sentences. This must be harmful to the automatic evaluation by RIBES. Accoding to this table, PostOrder gave higher RIBES scores to all systems and correlation between RIBES and Adequacy is lost as expected.

The ranking by RIBES-1.02.4 with "single" reference sentences completely agrees with Adequacy, but the weakest constraint, "postOrder", disagrees. Spearman's ρ of the two ranks is 0.800 but Pearson's r is as low as 0.256. It generates too many incomprehensible/misleading word orders, and they also raise RIBES scores of bad translations. On the other hand, "*proposed*" and "*caseMarkers*" agree with Adequacy except the ranks of tsbmt and the baseline Moses.

4 Concluding Remarks

RIBES is now widely used in Japanese-related translation evaluation. But RIBES sometimes penalizes good sentences because it does not regard scrambling. Once we have correct dependency trees of reference sentences, we can automatically enumerate semantically equivalent word orders. Less constrained reordering tend to generate syntactically ambiguous sentences. They become incomprehensible or misleading sentences. In order to avoid them, we introduced Simple Case Marker Constraint and restricted permutations to contiguous case marker children of verbs/ adjectives. Then, sentence-level correlation coefficients were improved.

The proposed enumeration method is also applicable to other automatic evaluation methods such as BLEU, IMPACT, and ROUGE-L, but we have to modify their codes for variable numbers of multi-reference sentences. We will examine them in the full paper.

We hope our method is also useful for other languages that have scrambling.

Acknowledgement

This research was supported by NTT Communication Science Laboratories.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for MT evaluation with improved correlation with human judgements. In *Proc. of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and Summarization*, pages 65–72.

Han Dan, Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2012. Head finalization reordering for Chinese-to-Japanese machine translation. In *Proceedings of SSST-6, Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 57–66.

Hiroshi Echizen-ya and Kenji Araki. 2007. Automatic evaluation of machine translation based on recursive acquisition of an intuitive common parts continuum. In *MT Summit XI*, pages 151–158.

Atsushi Fujii, Masao Uchimura, Mikio Yamamoto, and Takehito Usturo. 2008. Overview of the patent machine translation task at the NTCIR-7 workshop. In *Working Notes of the NTCIR Workshop Meeting (NTCIR)*.

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 *Working Notes of the NTCIR Workshop Meeting (NTCIR)*.

Isao Goto, Masao Utiyama, and Eiichiro Sumita. 2012. Post-ordering by parsing for japanese-english statistical machine translation. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 311–316.

Isao Goto, Ka Po Chow, Bin Lu, Eiichiro Sumita, and Benjamin K. Tsou. 2013. Overview of the patent machine translation task at the NTCIR-10 workshop. In *Working Notes of the NTCIR Workshop Meeting (NTCIR)*.

Katsuhiko Hayashi, Katsuhito Sudoh, Hajime Tsukada, Jun Suzuki, and Masaaki Nagata. 2013. Shift-reduce word reordering for machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1382–1386.

Tsutomu Hirao, Hideki Isozaki, Kevin Duh, Katsuhito Sudoh, Hajime Tsukada, and Masaaki Nagao. 2011. RIBES: An automatic evaluation method of translation based on rank correlation (in Japanese). In *Proc. of the Annual Meeting of the Association for Natural Language Processing*, pages 1115–1118.

Satoru Ikehara, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentaro Ogura, Yoshifumi Ooyama, and Yoshihiko Hayashi. 1997. *Goi-Taikei — A Japanese Lexicon (in Japanese)*. Iwanami Shoten.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, Hajime Tsukada, and Masaaki Nagata. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 944–952.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of translation quality using longest common subsequences and skip-bigram statistics. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 605–612.

Graham Neubig, Taro Watanabe, and Shinsuke Mori. 2012. Inducing a discriminative parser to optimize machine translation reordering. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 843–853.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 311–318.

Akihiro Tamura, Hiroya Takamura, and Manabu Okumura. 2007. Japanese dependency analysis using the ancestor-descendant relation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 600–609.

Xianchao Wu, Takuya Matsuzaki, and Jun'ichi Tsujii. 2012. Akamon: An open source toolkit for tree/forest-based statistical machine translation. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 127–132.