Dependency Analysis of Scrambled References for Better Evaluation of Japanese Translations

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

Abstract

In English-to-Japanese translation, BLEU (Papineni et al., 2002), the de facto standard evaluation metric for machine translation (MT), has very weak correlation with human judgments (Goto et al., 2011; Goto et al., 2013). Therefore, RIBES (Isozaki et al., 2010; Hirao et al., 2014) was proposed. RIBES measures similarity of the word order of a machine-translated sentence and that of a corresponding human-translated reference sentence.

RIBES has much stronger correlation than BLEU but most Japanese sentences have alternative word orders (scrambling), and one reference sentence is not sufficient for fair evaluation. Isozaki et al. (2014) proposed a solution to this problem. This solution generates semantically equivalent word orders of reference sentences. Automatically generated word orders are sometimes incomprehensible or misleading, and they introduced a heuristic rule that filters out such bad sentences. However, their rule is too conservative and generated alternative word orders for only 30% of reference sentences.

In this paper, we present a rule-free method that uses a dependency parser to check scrambled sentences and generated alternatives for 80% of sentences. The experimental results show that our method improves *sentence-level* correlation with human judgments. In addition, strong *system-level* correlation of single reference RIBES is not damaged very much.

We expect this method can be applied to other languages such as German, Korean, Turkish, Hindi, etc.

*This work was done while the second author was a graduate student of Okayama Prefectural University.

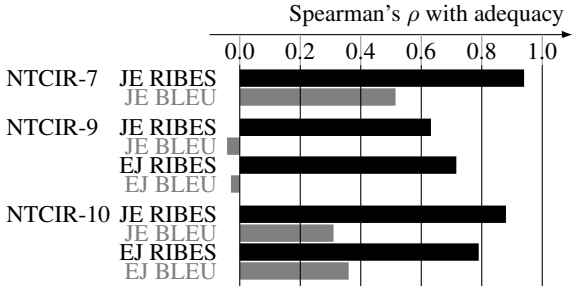


Figure 1: RIBES has better correlation with adequacy than BLEU (**system-level** correlation)

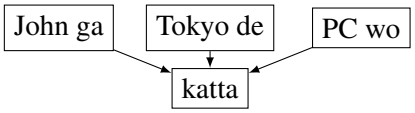
1 Introduction

For translation among European languages, BLEU (Papineni et al., 2002) has strong correlation with human judgments and almost all MT papers use BLEU for evaluation of translation quality. However, BLEU has very weak correlation with human judgments in English-to-Japanese/Japanese-to-English translation, and a new metric RIBES (Isozaki et al., 2010; Hirao et al., 2014) has strong correlation with human judgments. RIBES measures similarity of the word order of a machine translated sentence and that of a human-translated reference sentence. Figure 1 compares RIBES and BLEU in terms of Spearman's ρ with human judgments of adequacy based on NTCIR-7/9/10 data (Isozaki et al., 2010; Goto et al., 2011; Goto et al., 2013).

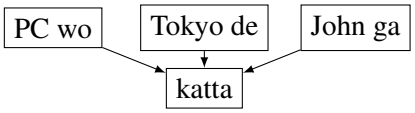
Japanese and English have completely different word order, and phrase-based SMT systems tend to output bad word orders. RIBES correctly points out their word order problems.

In this paper, we propose a method to improve "**sentence-level** correlation", which is useful for MT developers to find problems of their MT systems. If the sentence-level correlation is strong, low RIBES scores indicate bad translations, and we will find typical failure patterns from them.

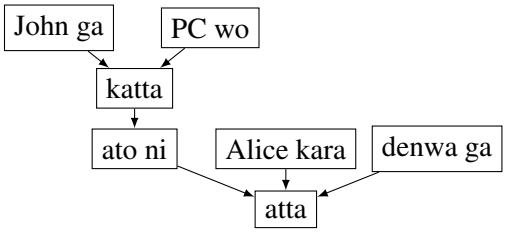
Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 450–456,
Lisboa, Portugal, 17-18 September 2015.



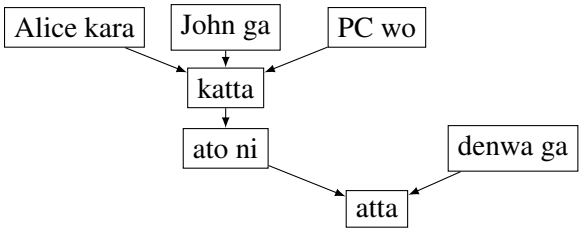
(a) "John ga Tokyo de PC wo katta"



(b) "PC wo Tokyo de John ga katta"



(c) "John ga PC wo katta ato ni Alice kara denwa ga atta"



(d) "Alice kara John ga PC wo katta ato ni denwa ga atta"

Figure 2: Dependency trees

However, improvement of **sentence-level** correlation is more difficult than **system-level** correlation and current automatic evaluation metrics do not have strong correlation. (Leusch et al., 2003; Stanojević and Sima'an, 2014; Echizen-ya and Araki, 2010; Callison-Burch et al., 2012)

1.1 Scrambling

As for Japanese translation, however, we should consider "scrambling" or acceptable reordering of phrases. For example, "John ga Tokyo de PC wo katta" (John bought a PC in Tokyo) consists of the main verb "katta" (bought) and its modifiers. "Ga", "de", and "wo" are case markers.

- "Ga" is a nominative case marker.
- "De" is a locative case marker.
- "Wo" is an accusative case marker.

This sentence can be reordered as follows.

1. John ga Tokyo de PC wo katta . (1.00)
2. John ga PC wo Tokyo de katta . (0.86)
3. Tokyo de John ga PC wo katta . (0.86)
4. Tokyo de PC wo John ga katta . (0.71)
5. PC wo John ga Tokyo de katta . (0.71)
6. PC wo Tokyo de John ga katta . (0.57)

All of the above sentences are acceptable and have the same meaning, and this is called "*scrambling*". However, RIBES outputs different scores for these sentences. When we use the first one as the reference sentence, RIBES output scores in the parentheses. Human judges will give almost equal scores to all of them, and we should improve these RIBES scores for better evaluation.

Scrambling is also observed in other languages such as German (Maier et al., 2014), Korean (Chun, 2013), Turkish (ldız et al., 2014), Hindi (Sharma and Paul, 2014), etc.

Figure 2 (a) shows the dependency tree of "John ga Tokyo de PC wo katta". Each box indicates a *bunsetsu* (chunk). Arrows indicate modification relations. The source node of an arrow modifies the target node of the arrow. The root "katta" has three modifiers (children), "John ga", "Tokyo de", and "PC wo". We can generate 3! = 6 word orders by post-order traversal of this tree because the order of siblings does not matter. Figure 2 (b) shows a permutation and its dependency tree. In this case, all permutations are acceptable.

However, more complex dependency trees tend to generate misleading/incomprehensible sentences. Figure 2 (c) shows such a sentence: "John ga PC wo katta ato ni Alice kara denwa ga atta". (After John bought a PC, there was a phone call from Alice). "X ato ni Y" means "After X, Y". "Denwa" means "a phone call". "Atta" means "there was".

This tree has 2! × 3! = 12 post-order permutations. Some of them are misleading. For example, "Alice kara John ga PC wo katta ato ni denwa ga atta" sounds like "After John bought a PC from Alice, there was a phone call" because "Alice kara" (from Alice) precedes "katta" (bought). This sentence will have a dependency tree in Figure 2 (d).

1.2 Rule-based filtering of bad sentences

Isozaki et al. (2014) tried to solve the above problem by automatic generation of reordered sentences and use of a heuristic rule (constraint) to filter out bad sentences.

- Use a Japanese dependency parser to get dependency trees of reference sentences.
- Check the dependency trees and manually correct wrong ones because sentence-level accuracy of dependency analyzers are still

low.

- In order to get Japanese-like head final sentences, output words in the corrected dependency tree in post-order. That is, recursively output all child nodes before a mother node. They called this method "postOrder".
- The above "postOrder" generates misleading/ incomprehensible sentences. In order to inhibit them, they introduced the following rule called "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.

  Here, we call this "rule2014".

This "rule2014" improved **sentence-level** correlation of NTCIR-7 EJ data. However, rule2014 is so conservative that only 30% of reference sentences obtained alternative word orders. In the next section, we present a method that covers more reference sentences.

2 Methodology

2.1 Our idea

We do not want to introduce more rules to cover more sentences. Instead we present a rule-free method. Our idea is simple: if a reordered sentence is misleading or incomprehensible, a dependency parser will output a dependency tree different from the original dependency tree. That is, use a dependency parser for detecting misleading sentences.

We apply a dependency parser to the reordered reference sentences. If the dependency parser outputs the same dependency tree with the original reference sentence except sibling orders, accept the word order as a new reference. Otherwise, it is a misleading word order and reject it. (We do not parse MT output because it is often broken and dependency analysis will fail.)

For example, "PC wo Tokyo de John ga katta" has the dependency tree in Figure 2 (b). This tree is the same as (a) except the order of three siblings. We don't care about the order of siblings, and accept this as a new reference sentence. On the other hand, the parser shows that "Alice kara John ga PC wo katta ato ni denwa ga atta" has the dependency tree in (d), which is different from (c) and we reject this sentence. We call this method "**compDep**" because it compares dependency trees of reordered reference sentences with the original dependency tree.

Each MT output sentence is evaluated by the best of RIBES scores for remaining reordered reference sentences. This is a sentence-level score. A system's score (system-level score) is the average of sentence-level scores of all test sentences.

2.2 Data and tools

We use NTCIR-7 PatentMT EJ data (Fujii et al., 2008) and NTCIR-9 PatentMT EJ data (Goto et al., 2011).[1] NTCIR-7 EJ human judgment data consists of 100 sentences $\times$ five MT systems. NTCIR-9 EJ human judgment data consists of 300 sentences $\times$ 17 MT systems. NTCIR provided **only one reference sentence** for each sentence. When we use only the provided reference sentences, we call it "single ref".

We apply a popular Japanese dependency parser CaboCha[2] to the reference sentences, and manually corrected its output just like Isozaki et al. (2014). 40% of NTCIR-7 dependency trees and 50% of NTCIR-9 dependency trees were corrected.

Based on the corrected dependency trees, we generate all post-order permutations. Then we apply CaboCha to these reordered sentences. We compare the dependency tree of the original reference sentence with that of a reordered reference sentence.

We accept a reordered reference sentence only when its tree is the same as that of the original reference sentence except the sibling order.

This tree comparison is implemented by removing word IDs and chunk IDs from the trees keeping their dependency structures and sorting children of each node by their surface strings. These *sorted* dependency trees are compared recursively from their roots.

3 Experimental Results

Table 1 shows that our compDep method succeeded in generating more reordered sentences (permutations) than rule2014. The column with #perms = 1 indicates failure of generation of reordered sentences. As for NTCIR-7, rule2014 failed for 70%

[1] NTCIR-8 did not provide human judgments. NTCIR-10 submission data was not publicly available yet at the time of writing this paper.

[2] http://code.google.com/p/cabocha/

NTCIR-7 EJ

#perms	1	2–10	11–100	101–1000	>1000	total
single ref	100	0	0	0	0	100
rule2014	70	30	0	0	0	100
compDep	**20**	61	15	4	0	100
postOrder	1	41	41	13	4	100

NTCIR-9 EJ

#perms	1	2–10	11–100	101–1000	>1000	total
single ref	300	0	0	0	0	300
rule2014	267	25	7	1	0	300
compDep	**41**	189	63	5	2	300
postOrder	0	100	124	58	18	300

Table 1: Distribution of the number of generated permutations (#perms=1 indicates the number of sentences for which the method didn't generate alternative word orders)

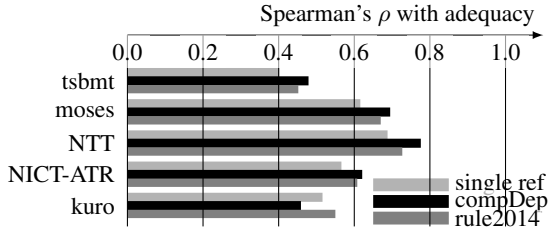


Figure 3: Improvement of **sentence-level** correlation with adequacy (NTCIR-7 EJ)

of reference sentences while compDep failed for only 20%. As for NTCIR-9, rule2014 failed for 89% (267/300) while compDep failed for only 14% (41/300).

From the viewpoint of the number of such failures, postOrder (§1.2) is the best method, but postOrder does not filter out bad sentences, and it leads to the loss of **system-level** correlation with adequacy (See §3.2).

3.1 Sentence-level correlation

Here, we focus on adequacy because it is easy to generate fluent sentences if we disregard adequacy. Figure 3 shows NTCIR-7 EJ results. our compDep succeeded in improving **sentence-level** correlation with adequacy for four MT systems among five. The average of ρ was improved from single ref's 0.558 to 0.606.

Figure 4 shows NTCIR-9 EJ results. our compDep succeeded in improving **sentence-level** correlation of all 17 MT systems. The average of ρ was improved from single ref's 0.385 and rule2014's 0.396 to compDep's 0.420. The improvement from single ref to compDep is statistically significant with $p = 0.000015$ (two-sided sign test) for NTCIR-9 data. The improvement from rule2014 to compDep is also statistically significant with $p = 0.01273$.

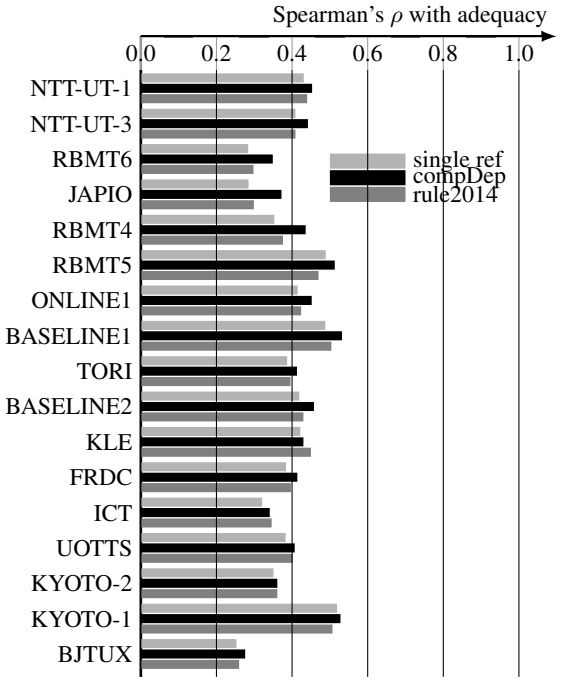


Figure 4: Improvement of **sentence-level** correlation with adequacy (NTCIR-9 EJ)

3.2 System-level correlation

Isozaki et al. (2014) pointed out that postOrder loses system-level correlation with adequacy because it also generates bad word orders.

Figure 5 shows that **system-level** correlation of compDep is comparable to that of single ref and rule2014. Spearman's ρ of compDep in NTCIR-7 (0.90) looks slightly worse than single ref and rule2014 (1.00). However, this is not a big problem because the NTCIR-7 correlation is based on only five systems as described in §2.2, and the NTCIR-9 correlation based on 17 systems did not degrade very much (compDep: 0.690, single ref: 0.695, rule2014: 0.668).

Table 2 shows details of **system-level** correlation of NTCIR-7 EJ. Single reference RIBES and rule2014 completely follows the order of adequacy. On the other hand, compDep slightly violates this order at the bottom of the table. NICT-ATR and kuro is swapped.

The "single ref" and "rule2014" scores of this table are slightly different from that of Table 5 of Isozaki et al. (2014). This difference is caused by the difference of normalization of punctuation symbols and full-width/half-width alphanumeric letters.

Figure 6 shows that the effects of manual correction of dependency trees. The average of sin-

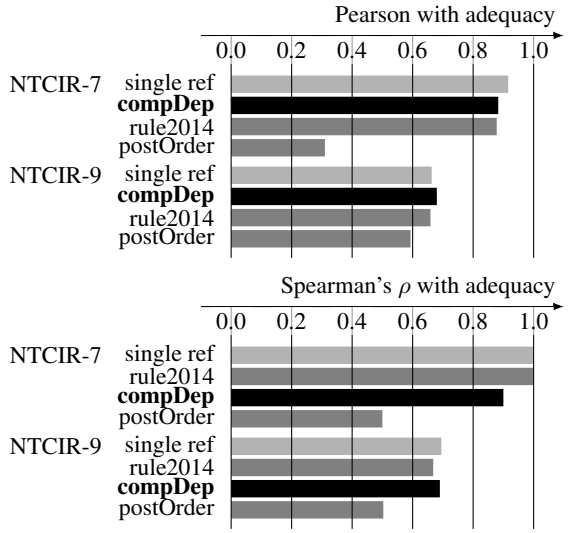


Figure 5: **System-level** correlation with adequacy

	Adequacy	Averaged RIBES		
		single ref	rule2014	compDep
tsbmt	3.527	0.722	0.726	0.750
Moses	2.897	0.707	0.720	0.745
NTT	2.740	0.670	0.682	0.722
NICT-ATR	2.587	0.658	0.667	0.706
kuro	2.420	0.633	0.643	0.711

Table 2: Details of **system-level** RIBES scores (NTCIR-7 EJ)

gle ref, compDep, and compDep without correction are 0.388, 0.422, and 0.420, respectively. Thus, the difference between compDep (with correction) and compDep without correction is very small and we can skip the manual correction step.

We used dependency analysis twice in the above method. First, we used it for generation of reordered reference sentences. Second, we used it for detecting misleading word orders.

In the first usage, we manually corrected dependency trees of the *given* reference sentences. In the second usage, however, we did not correct dependency trees of *reordered* reference sentences because some sentences have thousands of permutations (Table 1) and it is time-consuming to correct all of them manually. Moreover, some reordered sentences are meaningless or incomprehensible, and we cannot make their correct dependency trees. Therefore, we did not correct them. Our experimental results have shown that we can omit correction in the first step.

4 Related Work

Our method uses syntactic information. Use of syntactic information in MT evaluation is not a new idea.

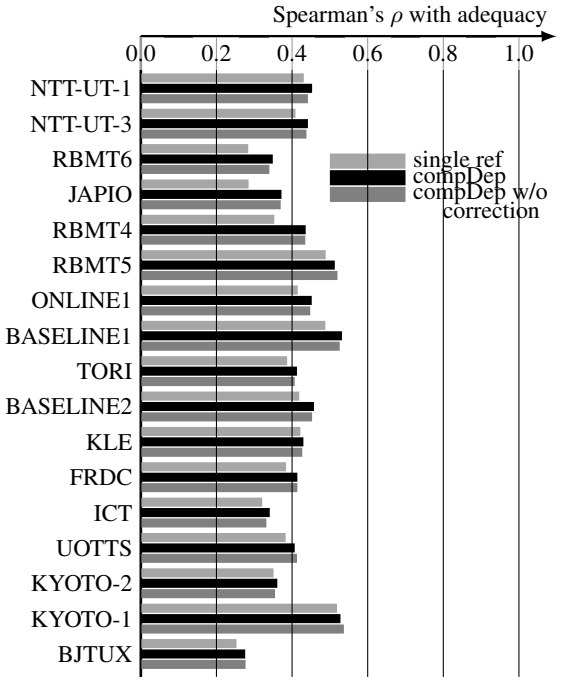


Figure 6: Effects of manual correction on compDep's correlation with adequacy (NTCIR-9 EJ)

Liu and Gildea (2005) compared parse trees of reference sentences and MT output sentences. They proposed four methods: STM, TKM, HWCM, DSTM, and DTKM. STM measures similarity by the number of matching subtrees. TKM uses Tree Kernel for the measurement. HWCM uses n-gram matches in dependency trees. DSTM and DTKM are dependency tree versions of STM and TKM respectively.

Owczarzak et al. (2007) used LFG-based typed dependency trees. They also introduced processing of paraphrases.

Chan and Ng (2008) proposed MAXSIM that is based on a bipartite graph matching algorithm and assigns different weights to matches. Dependency relation are used as a factor in this framework.

Zhu et al. (2010) proposed an SVM-based MT metric that uses different features in different granularities. Dependency relations are used as a feature in this framework.

We designed our method not to parse MT outputs because some MT outputs are broken and it is difficult to parse them. Our method does not parse MT outputs and we expect our method is more robust than these methods.

Recently, Yu et al. (2014) proposed RED, an evaluation metric based on reference dependency trees. They also avoided parsing of "results of

noisy machine translations" and used only dependency trees of reference sentences. However, their research motivation is completely different from ours. They did not mention *scrambling* at all, and they did not try to generate reordered reference sentences, but it is closely related to our method. It might be possible to make a better evaluation method by combining our method and their method.

Some readers might think that adequacy is not very reliable. WMT-2008 (Callison-Burch et al., 2008) gave up using adequacy as a human judgement score because of unreliability. NTCIR organizers used relative comparison to improve reliability of adequacy. The details are described in Appendix A of Goto et al. (2011).

5 Conclusions

RIBES (Isozaki et al., 2010) is a new evaluation metric of translation quality for distant language pairs. It compares the word order of an MT output sentence with that of a corresponding reference sentence. However, most Japanese sentences can be reordered and a single reference sentence is not sufficient for fair evaluation. Isozaki et al. (2014) proposed a rule-based method for this problem but it succeeded in generating alternative word orders for only 11–30% of reference sentences.

In this paper, we proposed a method that uses a dependency parser to detect misleading reordered sentences. Only when a reordered sentence has the same dependency tree with its original reference sentence except the order of siblings, we accept the reordered sentence as a new reference sentence. This method succeeded in generating alternative word orders for 80–89% and improved **sentence-level** correlation of RIBES with adequacy and its **system-level** correlation is comparable to the single reference RIBES.

In conventional MT evaluations, we have to prepare multiple references for better evaluation. This paper showed that we can automatically generate multiple references without much effort.

Future work includes use of the generated reference sentences in other metrics such as BLUE. We expect that this method is applicable to other languages such as German, Korean, Turkish, Hindi, etc. because they have scrambling.

References

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Futther meta-evaluation of machine translation. In *Proc. of the Workshop on Statistical Machine Translation*, pages 70–106.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proc. of the Workshop on Statistical Machine Translation*, pages 10–51.

Yee Seng Chan and Hwee Tou Ng. 2008. MAXSIM: A maximum similarity metric for machine translation evaluation. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 55–62.

Jihye Chun. 2013. Verb Cluster, Non-Projectivity, and Syntax-Topology Interface in Korean. In *Proc. of the Second International Conference on Dependency Linguistics*, pages 51–59.

Hiroshi Echizen-ya and Kenji Araki. 2010. Automatic evaluation method for machine translation using noun-phrase chunking. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 108–117.

Atsushi Fujii, Masao Utiyama, Mikio Yamamoto, and Takehito Utsuro. 2008. Overview of the patent translation task at the NTCIR-7 workshop. In *Working Notes of the NTCIR Workshop Meeting (NTCIR)*, pages 389–400.

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, 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)*.

Tsutomu Hirao, Hideki Isozaki, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2014. Evaluating translation quality with word order correlations (in Japanese). *Journal of Natural Language Processing*, 21(3):421–444.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 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.

Hideki Isozaki, Natsume Kouchi, and Tsutomu Hirao. 2014. Dependency-based automatic enumeration of semantically equivalent word orders for evaluating Japanese translations. In *Proc. of the Workshop on Statistical Machine Translation*, pages 287–292.

Olcay Taner Yıldız, Ercan Solak, Onur Görğ , and Razieh Ehsani. 2014. Constructing a Turkish-English Parallel TreeBank. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 112–117.

Gregor Leusch, Nicola Ueffing, and Hermann Ney. 2003. A novel string-to-string distance measure

with applications to machine translation evaluation. In *Machine Translation Summit*, pages 240–247.

Ding Liu and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation*, pages 25–32.

Wolfgang Maier, Miriam Kaeshammer, Peter Baumann, and Sandra Kubler. 2014. Discosuite - A parser test suite for German discontinuous structures. In *Proc. of the Language Resources and Evaluation Conference (LREC)*, pages 2905–2912.

Karolina Owczarzak, Josef van Genabith, and Andy Way. 2007. Dependency-based automatic evaluation for machine translation. In *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation,*, pages 80–87.

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.

Rahul Sharma and Soma Paul. 2014. A hybrid approach for automatic clause boundary identification in hindi. In *Proc. of the 5th Workshop on South and Southeast Asian NLP*, pages 43–49.

Miloš Stanojević and Khalil Sima'an. 2014. Fitting sentence level translation evaluation with many dense features. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 202–206.

Hui Yu, Xiaofeng Wu, Jun Xie Wenbin Jiang, Qun Liu, and Shouxun Lin. 2014. RED: A Reference Dependency Based MT Evaluation Metric. In *Proc. of the International Conference on Computational Linguistics (COLING)*, pages 2042–2051.

Junguo Zhu, Muyun Yang, Bo Wang, Sheng Li, and Tiejun Zhao. 2010. All in Strings: a Powerful String-based Automatic MT Evaluation Metric with Multiple Granularities. In *Proc. of the International Conference on Computational Linguistics (COLING)*, pages 1533–1540.