Enhancing Function Word Translation with Syntax-Based Statistical Post-Editing

John Richardson john@nlp.ist.i.kyoto-u.ac.jp
Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
Toshiaki Nakazawa nakazawa@pa.jst.jp
Japan Science and Technology Agency, Kawaguchi-shi, Saitama 332-0012, Japan
Sadao Kurohashi kuro@i.kyoto-u.ac.jp
Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan

Abstract

The generation of precise and comprehensible translations is still a challenge in the patent and scientific domain. In particular, function words are often poorly translated in standard machine translation systems, particularly across language pairs with greatly differing syntax. In this paper we exploit the target-side structure in tree-to-tree machine translation to post-edit function words automatically using a tree-based function word language model. We show that a significant improvement in human evaluation can be achieved with our proposed method.

1 Introduction

A high level of machine translation fluency is sought after in all subject domains. Translations with high adequacy however are especially important in patent and scientific translation, where it is particularly necessary to preserve the meaning of the input sentence in the generated translation.

Error analysis of state-of-the-art machine translation systems has shown that poorly translated function words are a major cause of loss in translation comprehensibility. For example, negation and passive structures can completely reverse their meaning when missing the correct function words, and it is important for understanding to select appropriate prepositions. We have also found that lack of (or incorrectly placed) relative pronouns has a large effect on preserving sentence meaning, and that badly formed punctuation impedes understanding.

Surprisingly few studies have been made specifically on improving function word translation for statistical machine translation systems, despite this having been looked at in rule-based systems (Arnold and Sadler, 1991). While we were unable to find any previous work on function word statistical post-editing, function words have been used to generate translation rules (Wu et al., 2011). The most similar approach to our method of editing function words used structural templates and was proposed for SMT (Menezes and Quirk, 2008). Statistical post-editing of MT output in a more general sense (Simard et al., 2007) and learning post-editing rules based on common errors (Elming, 2006; Huang et al., 2010) have shown promising results. The majority of statistical post-editing methods work directly with string output, however a syntactically motivated approach has been tried for post-editing verb-noun valency (Rosa et al., 2013).

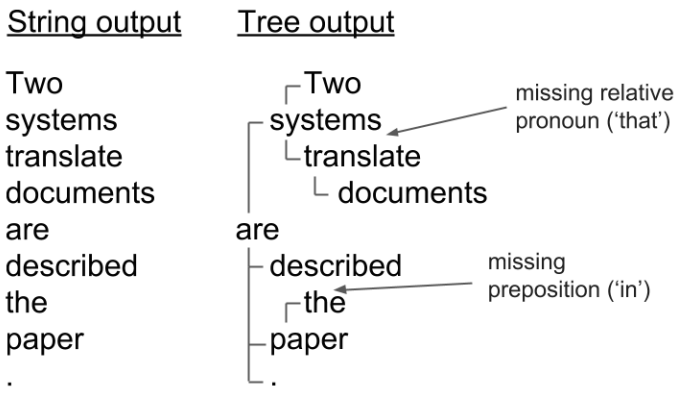

Figure 1: String vs Tree Output: The intended meaning of the translation is often unclear from string output. In this case we cannot tell easily that 'translate documents' is a relative clause (missing the relative pronoun 'which' or 'that') and that 'the paper' is a prepositional phrase (missing the preposition 'in') rather than the direct object of 'described'.

We believe that the intended meaning of a sentence is often unclear from flat MT output. For example, in Figure 1, the intended meaning is much clearer from the dependency tree representation. Based on this observation, we attempt to exploit the target structure of the output of a dependency tree-to-tree machine translation system in order to understand better the cause of the function word errors and therefore correct them more effectively.

2 Syntax-Based Post-Editing

Our proposed model starts with the dependency tree output of a tree-to-tree machine translation system. From this we analyze the position of function words and attempt to modify them with a tree-based function word language model.

We assume a set of function words F, a subset of the entire target-side vocabulary. We also define an empty token '<none>' which represents the lack of a function word. A root node and leaf nodes can be added to the tree to allow insertion of function words as the sentence root and leaves respectively.

A dependency tree can be decomposed into token–head pairs (t, t'). We derive a simple language model $P(f \mid t, t')$ approximating the probability of function word $f \in F$ being inserted between t and t'. The model is estimated over the training data by counting the occurence of (f, t, t') tuples where f is a function word appearing between t and t'. Note that to make this definition well-defined, we strictly require that function words have only one child. The probability $P(f \mid t, t')$ is then calculated as:

$$P(f \mid t, t') = \frac{count(f, t, t')}{\sum_{g \in F \cup <none>} count(g, t, t')} \quad (1)$$

In our experiments we include part-of-speech tags inside tokens to reduce homonym ambiguity (e.g. use 'set-NN' instead of 'set'). We also split $P(f \mid t, t')$ into two cases, $P_{left}(f \mid t, t')$ and $P_{right}(f \mid t, t')$, to consider the difference between t being a left or

right descendent of t'. We will write P_s to refer to whichever of P_{left} or P_{right} applies in each case.

2.1 Operations

For a token–head pair (t, t'), word insertion is performed when $P_s(f \mid t, t') > P_s(< none >\mid t, t')$ for some function word f. We choose the function word with the highest probability if there are multiple candidates. Replacement of function word t is performed similarly if $P_s(child(t) \mid f, t') > P_s(child(t) \mid t, t')$ for some other function word f. Similarly we choose the best f if there are multiple candidates. Deletion can be performed using the same method as for replacement by adding the function word '<none>' to F. The full algorithm for post-editing a tree $Tree$ is shown in Algorithm 1.

Algorithm 1 Post-Edit Tree

```
1: procedure PostEdit(Tree)
2:     loop:
3:     # Traverse tree from left-to-right
4:     for (t, t') ∈ Tree do
5:         if t ∈ F then
6:             child ← GetUniqueChild(t)
7:             # Find the best function word to replace t
8:             max_f, max_p ← t, P_s(t | child, t')
9:             for f ∈ F ∪ {< none >} do
10:                if P_s(f | child, t') > max_p then
11:                    max_f, max_p ← f, P_s(f | child, t')
12:                end if
13:            end for
14:            if max_f ≠ t then
15:                # Replace t with max_f and restart for entire tree
16:                Tree.Replace(max_f, child, t')
17:                goto loop
18:            end if
19:        else
20:            max_f, max_p ← t, P_s(< none >| t, t')
21:            # Find the best function word to insert
22:            for f ∈ F do
23:                if P_s(f | t, t') > max_p then
24:                    max_f, max_p ← f, P_s(f | t, t')
25:                end if
26:            end for
27:            if max_f ≠ < none > then
28:                # Add function word max_f and restart for entire tree
29:                Tree.Add(max_f, t, t')
30:                goto loop
31:            end if
32:        end if
33:    end for
34: end procedure
```

2.2 Filtering Replacements/Deletions with Word Alignments

In the majority of cases we found it counter-productive to replace or delete function words corresponding directly to non-trivial source words in the input sentence. For example, in a Chinese–English translation task, consider the two translations:

- 听/音乐 (listen/music) → listen *to* music
- 下面/100/米 (below/100/m) → 100m *below*

In the first sentence, the function word 'to' in the English translation has no corresponding word in the Chinese input and therefore its existence is based only on the target language model. In contrast, the preposition 'below' in the second sentence directly corresponds to '下面 (below)' in the input and care should be taken not to delete it (or change it to a completely different preposition such as 'above').

We therefore propose restricting replacement/deletion to function words that are aligned to trivial or ambiguous source-side words (function words without concrete meaning, whitespace, punctuation). This allows us to change for instance the unaligned 'to' in 'listen to' but not 'below' with an input alignment. The source–target word alignments are stored in the translation examples used by the baseline SMT system and kept track of during decoding.

3 Experiments

3.1 Data and Settings

We performed translation experiments on the Asian Scientific Paper Excerpt Corpus (ASPEC)[1] for Japanese–English translation. The data was split into 3 million training sentences, 1790 development sentences and 1812 test sentences.

We defined English function words as those tokens with POS tags of functional types such as determinants and prepositions, and treated Japanese particles as function words for the purposes of alignment-based filtering. The primary post-editing model was trained on the training fold of the ASPEC data. Since our model only requires monolingual data, for comparison we also trained a separate model on a larger (30M sentences) in-house monolingual corpus (Mono) of technical/scientific documents.

For the baseline SMT system we used KyotoEBMT (Richardson et al., 2014), a state-of-the-art dependency tree-to-tree translation system that can keep track of the input–output word alignments. Post-editing was performed on the top-1 translation produced by the tree-to-tree baseline system.

Japanese segmentation and parsing were performed with Juman and KNP (Kawahara and Kurohashi, 2006). For English we used NLParser (Charniak and Johnson, 2005), converted to dependency parses with an in-house tool. Alignment was performed with Nile (Riesa et al., 2011) and an in-house alignment tool. We used a 5-gram language model with modified Kneser-Ney smoothing built with KenLM (Heafield, 2011).

3.2 Evaluation

Human evaluation was conducted to evaluate directly the change in translation quality of function words. We found that automatic evaluation metrics such as BLEU (Papineni et al., 2002) were not sufficiently sensitive to changes (the change rate is relatively low for post-editing tasks) and did not accurately measure the function word accuracy.

In human evaluation we asked two native speakers of the target language (English) with knowledge of the source language (Japanese) to decide if the system output was

[1] http://lotus.kuee.kyoto-u.ac.jp/ASPEC/

better, worse, or neutral compared to the baseline. A random sample of 20 edited sentences were selected for each experiment and the identity of the systems was hidden from the raters. The Fleiss' kappa inter-annotator agreement (Fleiss et al., 1981) for wins/losses was 0.663, and when including neutral results this was reduced to 0.285.

3.3 Tuning and Test Experiments

We first perfomed a preliminary tuning experiment on the development fold of ASPEC to investigate the effect of model parameters. The results in Table 1 show for each row the model settings, the number of wins (+), losses (–) and neutral (?) results compared to the baseline, and the change rate (CR) over the entire development set.

The first three settings ('OnlyIns', 'OnlyRep', 'OnlyDel') show the effects of allowing only insertions, replacements and deletions respectively without using source–target alignments (see Section 2.2). We can see that the quality for deletions is lower than insertions and replacements, and error analysis showed that the major cause was deletion of function words aligned to content words in the input.

We reran the experiments using the alignment-based filtering ('AlignA' and 'AlignB') and found the results improved. While possible to achieve a higher change rate by allowing all three operations, we could only achieve a slight increase in accuracy by disallowing replacements (the setting 'AlignB'). The difference was mainly due to alignment errors, which caused more serious problems for replacement as they were able to alter sentence meaning more severely.

The best settings in the tuning experiment ('AlignB') were used to conduct the final evaluation on the unseen test data from ASPEC. We also compared models trained on the ASPEC training fold and on our larger monolingual corpus. Table 2 shows the final evaluation results. The results on the test set show significant improvement on win/loss sentences at $p < 0.05$. There was no clear improvement gained by increasing the size of model training corpus, however the change rate could be improved by using more data.

	Insert	**Replace**	**Delete**	**Align**	+	–	?	**CR**
OnlyIns	Yes	No	No	No	10	6	4	2.3
OnlyRep	No	Yes	No	No	11	7	2	5.5
OnlyDel	No	No	Yes	No	7	8	5	8.6
AlignA	Yes	Yes	Yes	Yes	11	7	2	10.5
AlignB	Yes	No	Yes	Yes	11	4	2	3.3

Table 1: Results of tuning experiments on development set.

	Insert	**Replace**	**Delete**	**Align**	+	-	?	**CR**
ASPEC	Yes	No	Yes	Yes	12	5	3	2.3
Mono	Yes	No	Yes	Yes	11	5	4	4.1
Both	Yes	No	Yes	Yes	23	10	7	3.9

Table 2: Final evaluation results on unseen data.

4 Error Analysis and Conclusion

The experimental results show that in general our proposed method is effective at improving the comprehensibility of translations by correctly editing function words. Ta-

Input	**転倒予防が重視されるのは，大腿骨頚部骨折との因果関係にある。**
Baseline	*Of* fall prevention is emphasized is the causal relation with femoral neck fracture.
Proposed	Fall prevention is emphasized is the causal relation with femoral neck fracture.
Input	**今後は，難治性疾患 (...) へと期待が寄せられる。**
Baseline	In the future , the expectation is being placed *to* the treatment of the intractable disease (...).
Proposed	In the future , the expectation is being placed *on* the treatment of the intractable disease (...).

Table 3: Examples of improved translations after deleting and replacing incorrect function words.

Input	**特に，簡易比色計によるりん酸塩の測定（モリブデン法）では，(...)** 。
Baseline	Especially, *in* the measurement of phosphate by simple colorimeter (molybdenum method), (...).
Proposed	Especially, the measurement of phosphate by simple colorimeter (molybdenum method), (...).
Input	**(...) 小型個体 (...) の水揚げ量を (...) １５%以下に抑えることが勧告された。**
Basline	(...) it was recommended that (...) suppress fish catch of small individuals (...) *to* 0,15%.
Proposed	(...) it was recommended that (...) suppress fish catch of small individuals (...) 0,15%.

Table 4: Examples of worsened translations. The first example shows a case where an important function word is lost, and this example was fixed by using the source–target alignments. The second example shows an error caused by model sparsity.

ble 3 gives examples of improved translations and Table 4 shows examples of worsened translations.

We found that using source–target alignments was effective in avoiding errors such as the first example in Table 4, however there remained some trickier cases where the alignment information was not sufficient, for example when function words were null or incorrectly aligned. The remainder errors were primarily caused by incorrect parsing and sparsity issues. The second example in Table 4 shows such a sparsity error, which could perhaps be fixed by normalizing numerical values.

In this paper we have shown that target-side syntax can be used effectively to improve the quality of scientific domain machine translation through the automatic post-editing of function words. We have presented an algorithm for inserting/deleting/replacing function words based on a simple tree-based language model and demonstrated the effectiveness of using source–target alignments to improve accuracy. In the future we plan to extend the model to provide more robustness against parsing/alignment errors and experiment with other language pairs.

Acknowledgments

We would like to thank the reviewers for their detailed comments and suggestions. We are also grateful to Raj Dabre for his assistance in conducting the human evaluation.

References

Arnold, D. and Sadler, L. (1991). EuroTra: An assessment of the current state of the ECs MT Programme. In *Working Papers in Language Processing*.

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

Elming, J. (2006). Transformation-based corrections of rule-based MT. In *EAMT 11th Annual Conference*.

Fleiss, L., Levin, B., and Paik, M. C. (1981). The measurement of interrater agreement. In *Statistical methods for rates and proportions (2nd ed)*, pages 212–236. Wiley.

Heafield, K. (2011). KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*.

Huang, A., Kuo, T., Lai, Y., and Lin, S. (2010). Discovering correction rules for auto editing. *International Journal of Computational Linguistics and Chinese Language Processing*, 15(3-4).

Kawahara, D. and Kurohashi, S. (2006). A fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 176–183. Association for Computational Linguistics.

Menezes, A. and Quirk, C. (2008). Syntactic models for structural word insertion and deletion during translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 735–744, Honolulu, Hawaii. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318. Association for Computational Linguistics.

Richardson, J., Cromières, F., Nakazawa, T., and Kurohashi, S. (2014). KyotoEBMT: An example-based dependency-to-dependency translation framework. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84, Baltimore, Maryland. Association for Computational Linguistics.

Riesa, J., Irvine, A., and Marcu, D. (2011). Feature-rich language-independent syntax-based alignment for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 497–507. Association for Computational Linguistics.

Rosa, R., Mareček, D., and Tamchyna, A. (2013). Deepfix: Statistical post-editing of statistical machine translation using deep syntactic analysis. In *Proceedings of the Student Research Workshop at the 51st Annual Meeting of the Association for Computational Linguistics*.

Simard, M., Goutte, C., and Isabelle, P. (2007). Statistical phrase-based post-editing. In *NAACL*.

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.