A Novel Rule Refinement Method for SMT through Simulated Post-Editing

Sitong Yang[1,2], Heng Yu[1,⋆], and Qun Liu[1,3]

[1] Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
[3] CNGL, School of Computing, Dublin City University
{yangsitong,yuheng,liuqun}@ict.ac.cn

Abstract. Post-editing has been successfully applied to correct the output of MT systems to generate better translation, but as a downstream task its positive feedback to MT has not been well studied. In this paper, we present a novel rule refinement method which uses Simulated Post-Editing (SiPE) to capture the errors made by the MT systems and generates refined translation rules. Our method is system-independent and doesn't entail any additional resources. Experimental results on large-scale data show a significant improvement over both phrase-based and syntax-based baselines.

1 Introduction

The quality of Statistical Machine Translation (SMT) is generally considered insufficient for use without a significant amount of human correction [1]. In the translation world, the term post-editing often refers to the process of manually correcting SMT output. While there does exist some documented cases of success, post-editing for SMT systems has not really become mainstream among professional translators, the main concerns are: unlike humans, the translation systems fail to learn from the post-editors corrections and keep making the same kind of mistakes.

One possible solution to this problem is by automatic post-editing [1–4]. Most of these works use another SMT system to capture the repetitive errors of the original system [2, 3], training a monolingual system to translate the original result into a better one. This method could arguably reduce the number of common errors and produce better results, but at the same time, it brings two side-effects: first, the flexibility: the training of the post-editing SMT system takes a long time, making it hard to adapt to different translation scenarios. Second, the pipeline of SMT systems involves several intermediate parts like alignment and rule-extraction, which will introduce additional errors and degrade the overall performance of the system.

⋆ Corresponding author.

C. Zong et al. (Eds.): NLPCC 2014, CCIS 496, pp. 113–122, 2014.

Another promising direction is to utilize post-editing results to capture the errors made by the SMT systems, and use a supervised error-driven paradigm to reinforce the original system. This method can make SMT systems more adaptive to all kinds of translation scenario without increasing the complexity of the system [5]. However, this task is challenging in two regards. First, the training data is expensive to generate, since it needs massive manual work for post-editing, Second, due to the poor quality of SMT output, it is hard to clearly identify the error and make the right correction.

In this paper, we follow the second direction and present a novel error-driven rule refinement method for SMT. First, we use a simulated post-editing paradigm in which either non-post-edited reference translation or manually post-edited translation from a similar MT system are used in lieu of human post-editorsi (Section 2). This paradigm allows us to efficiently collect the training data without expensive manual work and also enable the system to function in real-time post-editing scenarios without modification. Then we calculate the editing distance [6] between the translation output and the reference to capture the translation errors (Section 3.1), then generate refined rules based on the edit operations (Section 3.2). Finally, to ensure the goodness of the generated rules, we introduce a simple and effectively heuristic algorithm for rule-filtration (Section 3.3). We apply our method to both phrase-based and syntax-based SMT systems and gain an overall improvement of 1.4 BLEU point without using any additional resources. We also carry out experiment on multiple domains and find that our method works well on both news and medical domains (Section 4).

2 Simulated Post-Editing

In post-editing scenarios, humans continuously edit machine translation outputs into high quality translations, providing an additional, constant stream of data absent in batch translation. The data consists of highly domain revelant reference translations that are minimally different from MT outputs, making them ideal for learning. However, true post-editing data is infeasible to collect during system development and internal testing, as standard MT pipelines require tens of thousands of sentences to be translated with low latency. To address this problem, [8] formulated the task of simulated post-editing, wherein pregenerated reference translations are used as a stand-in for actual post-editing. This approximation is equivalent to the case where humans edit each translation hypothesis to be identical to the reference rather than simply correcting the MT output to be grammatical and meaning-equivalent to the source.

Our work uses this approximation for building large scale training set for refined rule-extraction. In our simulated post-editing task, we first use a baseline SMT system to translate all sentences in the bilingual corpus, then use the target side of bilingual corpus as the approximate post-editing results of the output from the SMT system. In this way, we are able to capture translation errors without any additional resource.

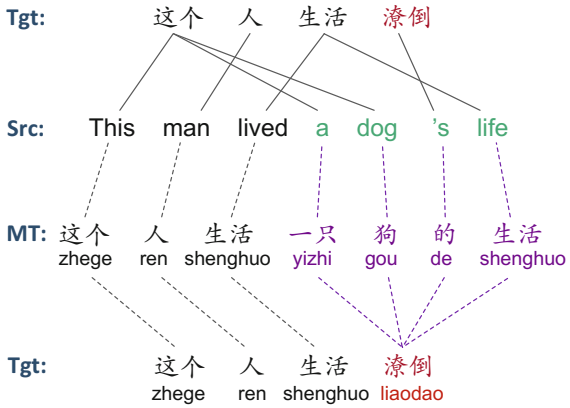

Fig. 1. An example of wrong alignment leading to bad rule-extraction. The alignment between the first and second line is the result of Giza++ [14]. ,the second alignment between "Src" and "MT" is done by MT decoding. The third alignment is generate by TER-plus [7].

3 Error-Driven Rule Refinement

The process of machine translation could be viewed as a search problem to find the best derivation of translation rules for the input sentence. So the quality of translation rules will greatly affect the performance of the system. On the other hand, the extraction of translation rules is based on word alignment of the bilingual corpus, which is mostly done in an unsupervised manner and the quality is not sufficient [14]. So the alignment error is generally considered as a bottle-neck for rule-extraction [10]. Figure 1 shows an example: the source side *a dog 's life* should be alignment to a specific Chinese term *liaodao*, but since *liaodao* is not a common translation for *dog*, the unsupervised alignment algorithm, i.e. Giza++ [14], will align *dog* to a more common word *zhege* and generate a false alignment between "Src" and "Tgt" in Figure 1. According to the alignment-consistent regulation [9], we will unable to extract the correct translation rule:

$$a\ dog\ `s\ life \longrightarrow liaodao$$

So in the real time translation, we may have to use a more common rule like:

$$a\ dog\ `s\ life \longrightarrow yizhi\ gou\ de\ shenghuo$$

This will lead to the wrong translation shown in the third line in Figure 1. These kind of errors are intrinsic for statistical models and hard to avoid. However, with the help of post-editing, we could easily identify those errors and get the right translation. So the motivation of our approach is to learn from the errors discovered in the post-editing process, then generate refined translation rules to correct the errors caused by statistical models. Our method consists of three key parts: error detection, rule extraction and rule filteration, we will give details in the following section.

3.1 Error Detection

Algorithm 1. Error-driven Rule Extraction

```
1: procedure TERp(Hypothesis h, Reference R)
2:     E ← ∞, Ops ← {}
3:     for r ∈ R do
4:         h′ ← h, e ← 0
5:         repeat
6:             Find operation s, that most reduces min-edit-distance(h, r)
7:             if s reduces edit distance then
8:                 h′ ← apply s to h′
9:                 e ← e + 1, p ← s
10:            end if
11:        until No operation that reduces edit distance remain
12:        e ← e+ min-edit-distance(h, r)
13:        if e < E then
14:            E ← e
15:            Ops ← p
16:        end if
17:    end for
18:    return E, Ops
19: end procedure
20:
21: procedure Rule Extraction(Operations ops)
22:    rule-set ← {}
23:    for p ∈ ops do
24:        t, t′ ← p
25:        s ← FindSource(t)
26:        rule-set ←< s, t′ >
27:    end for
28:    return rule-set
29: end procedure
```

We use the SiPE framework described in Section 2 to simulate the post-editing process and generate translation-reference pairs as "MT" and "Tgt" in Figure 1. To measure the absolute difference bewteen the two strings, we use Translation Error Rate Plus [7], which is an edit-distance based metric and an extenstion of TER [21] tailed for machine translation evaluation. Since the correct translations may differ not only in lexical choice but also in the order in which the words occur, TERp allows block movement of words, called shifts, within the hypothesis. Shifting a phrase is assumed to have the same edit cost as inserting, deleting or substituting a word, regardless of the number of words being shifted. This metric correlates well with the translation quality and could provide not only the score but also the edit operations needed to exactly transform the output into the reference. The pseudo-code is shown in the TERp procedure of Algorithm 1.

3.2 Rule Extraction

One advantage of TERp is that it generates an adequacy score by penalizing deletions, insertions, substitutions and shifts, This often allows it to calculates a set of shifts that largely align MT output to a reference, even when MT output uses significantly long orderings. This trait helps us to get larger chunk of modification rules rather than massive small pieces of rules which is hard to use in real time decoding. As shown in the last two lines in Figure 1, we apply TERp procedure on SMT output ("MT") and the reference ("Tgt"), and only one edit operation is needed:

$$yizhi\ gou\ de\ shenghuo \longrightarrow liaodao$$

Given the operation, we could perform rule extraction. The procedure straightforward: during decoding, we could align each source phrase s with its translation t, then using TERp procedure we could obtain the right modification t' of t. So it's easy to align the source side s (*a dog 's life* in Figure 1) with the correct translation t' (*liaodao*), generating the correct translation rule. Shown in Rule-extraction procedure in Algorithm 1.

However, the pending problem is that the translation probability of the new rule is hard to estimate. Since it would be very expensive and time consuming to modify the alignment and re-calculate the probability over the whole training-set, we propose two schemes for probability estimation:

First we could heuristicly set a high probability, assuming that all the newly learned rules should be preferred in translation.

Due to the complexity of MT errors, the generated rules may not be of high quality, further more, manually-set score may break the overall balance of the model, resulting in new errors in other translation scenarios. So we introduce a more balanced scheme by treating both original rule and the new rule equally, which allows the other features in SMT such as language model to determine which rule to use in real-time decoding.

The experimental results show that the second scheme achieves better performance, and the first scheme in certain cases will hurt the system.

3.3 Context-Based Rule Filteration

Since the quality of SMT output is relatively poor, a large number of modification rules will be generated based on our method. But due to the complexity of translation errors, some bad rules could also be generated. To address this issue, we propose a simple but effective rule-filtering method which use rule context to determine the goodness of the modification rule. We define the context of the rule C by the number of identical surrounding words, and P denotes the number of words within the rule. So C ensures the stable context of rule and filter out rules with unfaithful translations, And P will filter out too long modification rules which are unlikely to be used in test-set. In our experiment, we heuristically adjust C and P to obtain the best quality modification rules. And the best performance is achieved by setting $C \geq 2$ *and* $P \leq 5$.

4 Experiement

4.1 System Preparation and Data

To testify the solidness of our method, We conduct Chinese-to-English translation experiments on two different domains: news domain and medical domain, the information of the corpus is shown in Table 1. For comparison, We introduce two baselines:

1. Moses: a state-of-art phrase-based SMT system [15], available online[1]. we use the standard 11 features, set beam size to 200, max-phrase-length to 7, and distortion limit to 6
2. Hiero: an in-house implementation of Hierarchical Phrase-Based (HPB) model [16]. we use basic 8 features, and set beam size to 300, max-phrase-length
to 7.

We word-aligned the training data using GIZA++ with refinement option "grow-diag-and" [17], and trained 4-gram language model on giga-xinhua corpus using the SRILM toolkit [20] with modified Kneser-Ney smoothing. For parameter tuning, we use minimum error-rate training [12] to maximize the Bleu score on the development set. We evaluate translation quality using case-insensitive Bleu-4, calculated by the script `mteval-v11b.pl`. We also report the TERp scores calculated by TER-Plus [7].

Table 1. Overview of the data-sets used in the experiment. All the numbers are sentence count.

Domain	Training-set	Dev-set	Test-set		
News	240k	Nist02	Nist04	Nist05	Nist06
Chemistry	560k	1000	1000		

4.2 Results and Analysis

We first show the results on news domain in table 2: "heuristic" and "balanced" denotes the two schemes for assigning translation probability to refined rules. Since "heuristic" assigned high probability to refined rules, they were always preferred in decoding, thus hurting the system by breaking the balance of the statistical model. On the other hand, "balanced" scheme assigned the same probability with the original rule, so in real time decoding, other SMT features like language model could determine the right rule to use. For phrase-based system, our method gains an average improvement of 1.42 bleu points over all test-sets. But for hirarchical phrase-based system, the improvement is relatively small. The reason is two-folded: first HPB model generates hirarchical rules which could

[1] `www.statmt.org/moses/`

Table 2. Final results on news domain, the number in bold means the improvement is statistically significant ($p < 0.05$)

System	Bleu				TERp			
	04	05	06	avg	04	05	06	avg
moses	32.02%	29.00%	27.18%	29.34%	61.47%	64.04%	66.45%	64.47%
heuristic	31.73%	28.66%	26.22%	28.87%	62.23%	64.67%	67.03%	64.64%
balanced	**33.47%**	**30.02%**	**28.80%**	**30.76%**	**60.24%**	**59.37%**	**63.89%**	**61.77%**
hiero	34.10%	29.89%	28.78%	30.92%	59.55%	62.73%	64.84%	62.37%
balanced	34.09%	29.87%	**28.81%**	30.92%	59.56%	62.75%	64.85%	62.38%

greatly expand the rule coverage, which compensate the effect of bad alignment. The second reason may be that the alignment quality of the news domain is relatively good with fewer rare word so the rule-extraction errors is not very severe.

We also show the TERp score in the last column, and it's reasonable that the performance is in accordance with bleu. The average drop is TERp score is about 2.7.

The results on medical domain is also promising, shown in Table 3. We can see that on phrase-based model the bleu gains is 0.51 point, at the same time, for hirachical phrase-based model the improvement is 0.78, much more significant than that on the news domain. This is because there are many formula and special terms in medical-domain corpus which makes it hard for unsupervised alignment, causing more errors in rule extraction. So our method produced more significant improvement by generating better translation rules.

Table 3. Performance on testset of medical domain

System	BLEU	TERp
moses	29.64%	66.06%
Ours	**30.15%**	**63.80%**
hiero	29.48%	63.53%
Ours	**30.26%**	**62.57%**

The effect of rule filteration is also critical to our approach. Since there are still some noise in the generated rules, we tried different heuristic filter settings to test the performance of the system. Figure 2 shows the results, we can see that more strict filteration settings produced better performance: adding 0.5 million new rules degrades the performance a little bit, then we gradually constrain the filteration settings and the performance gets better with the peak of 0.7 bleu point gain.

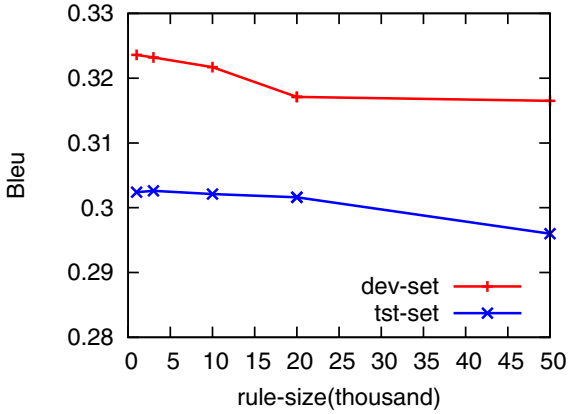

Fig. 2. The impact of different filteration heuristics on Bleu score

5 Related Work

Our work belongs to the family of Statistical Post-editing [1], which intends to use statistical method to capture the errors made by the translation system to further improvement performance. Simard et.al (2007) and Lagarda et.al (2009) both use a phrase-based SMT system to post-edit the output of a rule-based system, this method could combine the merits of both models and yield significant improvement. Bechara et.al (2011) is the first to directly use a SMT system to post-edit another SMT system, the key part of their success is that their post-editing system uses additional corpus to train. Our method is very different from theirs in that rather than relying on another powerful system, we try to dynamically improve the original system without bringing in more complex models and additional error. Besides, our method don't require any additional resources and learn directly from SMT training data.

There are also some work focus on utilize post-editing techniques to improve MT. Isabelle et.al (2007) use automatic post-editing to solve domain adaptation problem in MT. Mundt et.al (2012) learn to automatically recover dropped content words from post-editing. And Denkowski et.al (2014) use post-editing to train an online adaptation framework for SMT. Our work is in the same spirit with theirs, but we focus on rule refinement task.

The simulated post-editing paradigm in our work could also be viewed as a force decoding process [23, 24], in which we can boost new translation rules for better forced decoding. The difference is that we don't require a strict forced decoding, which is too strict for some MT cases, but try to detect errors in the process and generate refined rules.

6 Conclusion

In this paper we have introduced a novel rule refinement method for SMT. We use a simulated post-editing paradigm to efficiently collect the training data. And use TER-Plus for translation error detection and modification rule-extraction. Finally, to ensure the goodness of the generated rules, we introduce a simple and effectively heuristic algorithm for rule-filtration. We apply our method to both phrase-based and syntax-based SMT systems and gains an overall improvement of 1.4 BLEU point without using any additional resources. In the future, we will try to test our method on more complex translation models and produce more powerful feedbacks to improve SMT systems.

Acknowledgement. We thank the three anonymous reviewers for helpful suggestions. The authors were supported by CAS Action Plan for the Development of Western China (No. KGZD-EW-501) and National Natural Science Foundation of China (No. 2012BAH39B03). Qun Liu's work was partially supported by the Science Foundation Ireland (Grant No. 07/CE/I1142) as part of the CNGL at Dublin City University. The views and findings in this paper are those of the authors and are not endorsed by the Chinese governments.

References

1. Simard, M., Goutte, C., Isabelle, P.: Statistical phrase-based post-editing. In: Proceedings of NAACL (2007)
2. Bechara, H., Ma, Y., van Genabith, J.: Statistical post-editing for a statistical MT system. In: Proceedings of MT Summit XIII, pp. 308–315 (2011)
3. Lagarda, A.L., Alabau, V., Casacuberta, F., et al.: Statistical post-editing of a rule-based machine translation system. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion, vol. Short Papers, pp. 217–220. Association for Computational Linguistics (2009)
4. Dugast, L., Senellart, J., Koehn, P.: Statistical post-editing on SYSTRAN's rule-based translation system. In: Proceedings of the Second Workshop on Statistical Machine Translation, pp. 220–223. Association for Computational Linguistics (2007)
5. Denkowski, M., Dyer, C., Lavie, A.: Learning from Post-Editing: Online Model Adaptation for Statistical Machine Translation. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (2014)
6. Navarro, G.: A guided tour to approximate string matching. Journal of ACM computing surveys (CSUR) 33(1), 31–88 (2001)
7. Snover, M.G., Madnani, N., Dorr, B., et al.: TER-Plus: paraphrase, semantic, and alignment enhancements to Translation Edit Rate. Journal of Machine Translation 23(2-3), 117–127 (2009)
8. Hardt, D., Elming, J.: Incremental Re-training for Post-editing SMT. In: Proceedings of AMTA (2010)

9. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Jounral of Computational linguistics 29(1), 19–51 (2003)
10. Liu, Y., Xia, T., Xiao, X., et al.: Weighted alignment matrices for statistical machine translation. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, vol. 2, pp. 1017–1026. Association for Computational Linguistics (2009)
11. Brown, P.F., Cocke, J., Pietra, S.A.D., et al.: A statistical approach to machine translation. Journal of Computational linguistics 16(2), 79–85 (1990)
12. Och, F.J.: Minimum error rate training in statistical machine translation. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, vol. 1, pp. 160–167. Association for Computational Linguistics (2003)
13. Papineni, K., Roukos, S., Ward, T., et al.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics (2002)
14. Och, F.J., Ney, H.: The alignment template approach to statistical machine translation. Journal of Computational linguistics 30(4), 417–449 (2004)
15. 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., Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation. In: Annual Meeting of the Association for Computati onal Linguistics (ACL), demonstration session, Prague, Czech Republic (2007)
16. Chiang, D.: Hierarchical Phrase-Based Translation. Journal of Computational Linguistics 33(2), 201–228 (2007)
17. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proceedings of the 2003 Conference of the North Americanz Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL 2003), vol. 1, pp. 48–54. Association for Computational Linguistics, Stroudsburg (2003)
18. Mundt, J., Parton, K., McKeown, K.: Learning to Automatically Post-Edit Dropped Words in MT. In: Proceedings of AMTA (2012)
19. Isabelle, P., Goutte, C., Simard, M.: Domain adaptation of MT systems through automatic post-editing. In: Proceedings of MTS (2007)
20. Stolcke, A.: SRILM – An Extensible Language Modeling Toolkit. In: Proceedings of Intl. Conf. on Spoken Language Processing, Denver, vol. 2, pp. 901–904 (2007)
21. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of Association for Machine Translation in the Americas, pp. 223–231 (2006)
22. Niessen, S., Och, F., Leusch, G., Ney, H.: An evaluation tool for machine translation: fast evaluation for MT research. In: Proceedings of the 2nd International Conference on Language Resources and Evaluation, pp. 39–45 (2000)
23. Yu, H., Huang, L., Mi, H., Zhao, K.: Max-Violation Perceptron and Forced Decoding for Scalable MT Training. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1112–1123 (2013)
24. Liang, H., Zhang, M., Zhao, T.: Forced decoding for minimum error rate training in statistical machine translation. Journal of Computational Information Systems (8), 861868 (2012)