RGraph: Generating Reference Graphs for Better Machine Translation Evaluation

Hongjie Ji, Shujian Huang(✉), Qi Hou, Cunyan Yin, and Jiajun Chen

State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing, China
{jihj,huangsj,houq,yincy,chenjj}@nlp.nju.edu.cn

Abstract. Statistical machine translation systems perform parameter learning (i.e. training) basing on automatic translation evaluation methods, which usually evaluate the translation quality according to one or more human-translated references. Although producing more references would improve the coverage of translation choices and lead to improved training performances, only several references are used due to the cost of human translation. In this paper, we propose automatic methods to explore the information among the limited references. By generating a reference graph (RGraph) from given references, we could automatically generate exponential number of references. These diverse references make it possible to better evaluate each individual translations, without using any other resources. Experiments showed that our RGraph could improve the evaluation performance and lead to better tuned machine translation systems. The method could be extended to improve the evaluation with single reference as well.

Keywords: Machine Translation Evaluation · Reference graph · Parameter learning · Evaluation metrics

1 Introduction

Statistical machine translation systems usually need to learn the system parameters using some training/tuning algorithms, such as Minimum Error Rate Training (MERT), Pair-wise Ranking Optimization (PRO) [16,26]. These training algorithms adjust the parameters so that the output of the system has a high translation quality. Because the translation quality is hard to evaluate, early attempts usually employ human to make the quality decision, which is very expensive and time-consuming [17]. Various metrics have been proposed to use automatical evaluation methods to replace human decisions, such as Word Error Rate (WER), BLEU and Translation Edit Rate (TER) [28,31]. These methods automatically evaluate similarity (or distance) between machine translation outputs and human-translated references.

Current evaluation methods can be roughly divided into two categories. The first category of methods are based on sentence-matching. These methods usually compute an explicit one-to-one matching between words in the output sentence and a certain reference. The matching is then evaluated using metrics such as Levenshtein Distance [4,20,21,31,32] or matching precision and recall [4,11,12]. However, in order to generate an explicit matching, only one unique reference is considered. If there are multiple references, these methods need to select the one that is closest to the translation for the matching. This leads to potential problems when evaluating diverse translations. Because the same source language word or phrase could have multiple correct translations due to paraphrasing, the number of correct translations for a given source language sentence grows exponentially w.r.t the length of the sentence. When it is impossible to get an ideally "close" reference in one of the references, these methods may reach inaccurate quality decisions. In order to consider diversity in the evaluation process, external resources such as paraphrase tables or lexicon analysis are employed [1,4], but these resources may also bring in noises for the evaluation.

The second category of methods are segment-based. These approaches split the reference and the output into smaller segments, e.g. n-grams. The translation quality could be measured by the matching rate of segments, but without an explicit one-to-one matching [6,15,23,28]. Segment-based methods could naturally consider diverse translations, because the matching segments could come from any one of the references. However, because the matching of a translation segment could also come from any part of the reference, especially for those common function words, the quality measure could be less accurate. Besides, because the evaluation only focus on segments, the fluency of the whole sentence is less considered.

In this paper, we propose an evaluation approach that enjoys the benefits of both the above categories. By splitting the reference into segments and aligning them into a *Reference Graph*, we deeply explore the information inside the limited references and generate a compact representation that could represents exponentially many correct translations (Sect. 2). Selecting a path in the graph determines a certain reference translation; and metrics could be used to calculate distance between the output and the selected reference (Sect. 3). We also propose methods that explore monolingual resources to build the reference graph when only one reference is given (Sect. 2.3). Experiments demonstrate that our evaluation approach could achieve evaluation results better correlated to human decisions. Furthermore, tuning with reference graphs significantly improves the training performance of a large scale machine translation system, in both multiple reference and single reference settings (Sect. 4).

2 Constructing Reference Graph

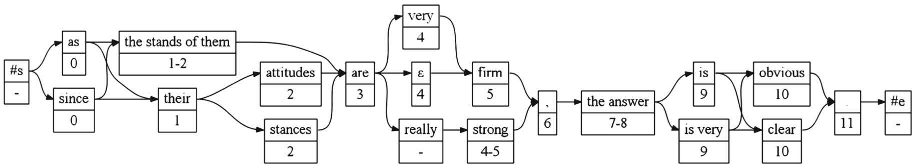
The major problem of representing possible translations of a given source sentence by independent references is that it is impossible to enumerate all possible translations. So the evaluation would be unfair to the sentences with different word orders (shown in Fig. 1). Inspired by the practice of using confusion networks or lattices to represent translations from different systems in system

Ref1: as the stands of them are firm , the answer is clear .
Ref2: since their standces are really strong , the answer is very obvious .
Ref3: as their attitudes are very firm , the answer is obvious .
Ref4: since the stands of them are really strong , the answer is obvious .

(a)

Tran1: since their attitudes are firm , the answer is very clear .
Tran2: as since the stands are really very obvious ,

(b)



(c)

Fig. 1. The motivation example. (a) Shows the original 4 references; (b) shows two different translations: Tran1 is not close to any of the reference, which may lead to a lower score; Tran2 has several suspicious function words in incorrect places, which may lead to an overrate; (c) shows the RGraph built from these references, which may contain proper references for evaluating the quality of translations. In each vertex of the RGraph, upper cell presents the translations and the lower cell presents the indexes of the corresponding source part.

combination [5,13,14,18,19,25], we propose to organize the references in a compact representation, named *Reference Graph* (RGraph) (shown in Fig. 1). With RGraph, the translation information inside the limited references can be better explored.

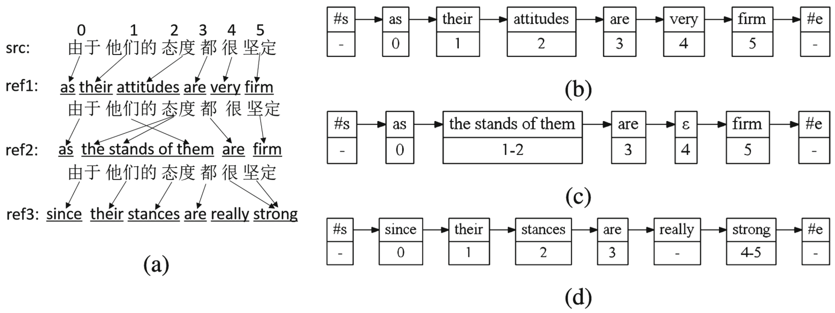
Different from the practice of confusion networks or lattices, which aligns all other translations to a selected back-bone translation and arrange them in the same order, we need to keep as many different word orders as possible to evaluate different translations. As a result, we choose to use the source sentence as the "back-bone" of the graph and align all possible translations in the reference to the source sentence. The alignment could be obtained using off-the-shelf alignment tools such as GIZA++ [27]. With the alignment, the construction of an RGraph is proceed in two steps: constructing sub-graph from every single reference, merging sub-graphs into the RGraph.

For convenience, we use the following notations throughout this paper. We denote the source sentence as $f = f_1, f_2, ..., f_l$, the reference as $r = r_1, r_2, ..., r_n$. The RGraph of f and its reference set R is a directed graph $D(f, R)$, or simply D, which consists of a vertex set V and an edge set E. Each vertex v in V represents a translation from source segment (f_i, f_j) to reference segment (r_p, r_q), including two pseudo vertices $\#_s$ and $\#_e$, which denotes the start and the end of a sentence. Because all references correspond to the same source sentence, we use source indexes instead of words in each node. So each vertex could be denoted as $\langle(i, j), (r_p, r_q)\rangle$. Each edge e in E connects two translation segments that are adjacent in the source side, e.g. from (i, k) to $(k + 1, j)$. A path p is made up of one or more vertexes connected by edges. p represents a source segment together with its translation, which is the concatenation of the source and target part of all vertexes in the path. Specifically, each path from $\#_s$ to $\#_e$ represents a possible translation of the whole sentence f. We use D_s, D_m and D_t to represent the directed graph constructed from a single reference, from monolingual resources and from the translation to be evaluated, respectively.

2.1 Constructing the Sub-graph

With a source sentence f, a reference r and the alignment A between them. We construct a sub-graph D_s that contains all translation information in the reference while maintaining the original order of r. The basic idea is to split the translation into minimum and monotonic blocks, with the reordering information covered inside each block. Specifically, the following three conditions should be satisfied: (1) no word in a block is aligned to words in other blocks; (2) the block is monotonic in both source and target side to its adjacent blocks; (3) the block is minimum, which means it could not be splitted into smaller blocks. An example of the block splitting results is in Fig. 2a.

The concept of blocks is the same as Mariño et al. [24], where similar bilingual n-grams are defined and referred to as *tuples*. With blocks, the whole sentence becomes a block sequence, which has the same order in both source and target side. This monotonic property leads to convenient algorithms for both constructing the graph and performing evaluation. The blocks are required to be minimum to cover the translation information at the finest level. Then larger segments of translation could be easily formed by merging consecutive blocks.



0 1 2 3 4 5
src: 由于 他们的 态度 都 很 坚定
ref1: as their attitudes are very firm
由于 他们的 态度 都 很 坚定
ref2: as the stands of them are firm
由于 他们的 态度 都 很 坚定
ref3: since their stances are really strong
(a)
#s - | as 0 | their 1 | attitudes 2 | are 3 | very 4 | firm 5 | #e -
(b)
#s - | as 0 | the stands of them 1-2 | are 3 | ε 4 | firm 5 | #e -
(c)
#s - | since 0 | their 1 | stances 2 | are 3 | really - | strong 4-5 | #e -
(d)

Fig. 2. An example of block splitting and sub-graph construction. (a) Shows blocks splitted according to the alignments for 3 references, respectively, where each underlined English phrase, together with its source counterpart forms a block; (b), (c) and (d) show sub-graphs constructed from ref1, ref2 and ref3, respectively, with one vertex to represent each block.

To construct the sub-graph, we generate a vertex for each block to represent the source segment (i, j), together with its target translation (r_p, r_q). These vertexes are then connected in their original order to form a directed sub-graph,

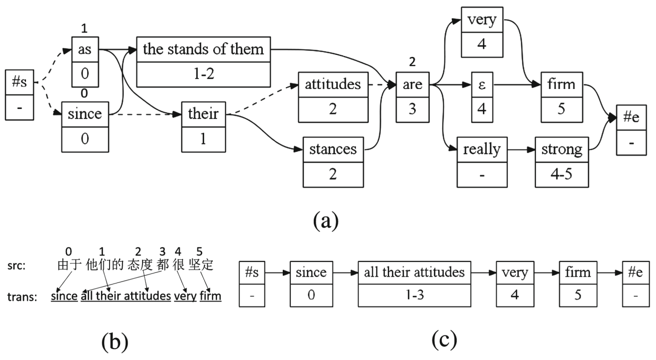


Fig. 3. An example of path-based graph search. The example searches in the RGraph in (a) for the closest reference for the translation in (b), which has a translation graph D_t shown in (c). We denote, for each vertex, the *minEdits* on top of it and the *minPath* to it by dashed lines. By far, the search proceeds at the *are* vertex. For the source span (1,3), the algorithm compares the following three paths "the stands of them are", "their attitudes are" and "their stances are", and selects the second one as the *minPath*. Further computation will be based on this selection.

which is the base of building a larger RGraph. Examples of the constructed sub-graphs are in Fig. 2.

Different from Mariño et al. [24] where NULL-aligned target words are appended to the previous words, we treat all NULL-aligned words as separate vertexes and insert them into the sub-graph as well. The ε vertex in Fig. 2c and the *really* vertex in Fig. 2d are examples in these cases.

2.2 Merging Sub-graphs

For multiple references, each reference will produce a sub-graph, these sub-graphs are then merged all together to form the RGraph. The merging process starts with an empty graph D and iteratively merges each sub-graph D_s into D. The merging first unions the vertex and edge set of the sub-graph into D, where vertexes covering the same source block with the same target translation are considered the same. Then new edges are added between adjacent vertexes in D and D_s. For example, between vertex A$\langle(i,k),(r_p,r_q)\rangle$ and B$\langle(k+1,j),(r_{p'},r_{q'})\rangle$ in D, an edge will be added from A to B if none of B's neighbors are NULL-aligned. It is easy to see that sub-graphs, which have just one path for the whole sentence, are simple versions of the RGraph. Figure 3a shows the result of merging three reference sub-graphs in Fig. 2.

2.3 Monolingual Extension of the Graph

In cases where there are only one single reference translation provided for each source sentence, it might be difficult to perform fair evaluations for diverse translation outputs. To solve the problem, we propose to enhance the single reference by building an RGraph using monolingual information, such as dictionaries, paraphrase tables, etc.

The single reference is firstly used to generate a sub-graph D_s as in previous sections. The monolingual resources are then used to generate alternative translations for each vertex or consecutive vertexes (i.e. paths) in D_s. We create new vertexes for these alternative translations and add them to an monolingual graph D_m. D_m could then be merged with D_s to generate the RGraph D as well, like in Sect. 2.2. Without sentence-level translations, it is not possible to add edges between vertexes in D_m. So we keep these vertexes unconnected. As a result, D_m could be seen as a special case of D with only vertexes but no edges.

3 Graph-Based Evaluation Metrics

Note that any given translation to be evaluated could also be transformed into a sub-graph (denoted by D_t) with the method described in Sect. 2.1. The alignment between the translation and the source sentence could be obtained by automatic alignments or as an extra output from the translation process. So the evaluation task now becomes evaluating the translation in D_t with RGraph D.

The evaluation could be executed in two steps. First, find the closest reference from the RGraph. Second, evaluate the translation according to the closest reference, with any metrics. The effectiveness of the two-step approach is evidenced by the experiment of Human-targeted Translation Error Rate, i.e. HTER [31], where human translator edits the reference so that it become closer to the given translation. Here we replace the human editing process with a graph matching process, which could be performed automatically without relying human editors.

To find the closest reference among the exponentially many possible references in the RGraph, we employ edit distance as the measure for closeness and perform a left-to-right search on the graph. Because the vertexes in D and D_t may cover different source segments, we use the paths as the basic search units.

The search process is similar to the Dijkstra's algorithm, except that our algorithm operates in the path level instead of vertex level. The distance is computed on-the-fly. As shown in Algorithm 1, for each vertex v, we compute the path ($minPath$) in D which ends in v, and has the number minimum of edits ($minEdits$). The algorithm starts from the path with a single node $\#_s$. It searches all possible paths according to the ending index of their source side block (using the priority queue, line 2), so that before calculating the edit distance for path p, all its prefix paths have been computed. Every time a path p fits a given path p_t in D_t suggests that the two paths cover the same source segment (line 7), thus their target sides could be measured by the edit distance algorithm (line 8). For any vertex v, only the path with minimum edits will be recorded for further extension (ensured by condition in line 9). Following computation would start from the successor of p, taking the previous path p and its minimum edits as basis (lines 12–13). If current path doesn't fit any path in D_t, all its successor will be enumerated to extend the path (lines 15–17). In worst cases, the extension lasts until the end word $\#_e$.

Algorithm 1. Path-based Graph Search Algorithm

```
Input: translation graph $D_t$ with $V_t$, $E_t$ and RGraph $D$ with $V$, $E$
1: initialize each vertex $v \in V$ with $v.minEdits \leftarrow$MAX, $v.minPath \leftarrow \emptyset$
2: priority queue of paths $pq \leftarrow \emptyset$  ▷ with the ending index of the source side block as the priority
3: path $start \leftarrow \#_s$, $start.edits \leftarrow 0$
4: $pq$.push($start$, 0)
5: while $pq \neq \emptyset$ do
6:     path $p \leftarrow pq$.pop(), $v \leftarrow$ the last vertex in $p$
7:     if $p$ fits a path $p_t$ in $D_t$ then          ▷ $p$ and $p_t$ cover the same source segment
8:         $p.edits \leftarrow p.edits+$ editDistance($p$, $p_t$)
9:         if $p.edits < v.minEdits$ then
10:            $v.minEdits \leftarrow p.edits$, $v.minPath \leftarrow p$
11:            for all successor $v' \in V$ of $p$ do          ▷ finding shortest paths starting from $v'$
12:                path $newp \leftarrow v'$, $newp.edits \leftarrow p.edits$, $newp.prePath \leftarrow p$
13:                $pq$.push($newp$, $v'.end$)          ▷ $v'$ covers source block ($start, end$)
14:    else          ▷ extending $p$ with following vertexes to get a fit with $D_t$
15:        for all successor $v' \in V$ of $p$ do
16:            path $newp \leftarrow p + v'$, $newp.edits \leftarrow p.edits$ ▷ $p + v'$ means appending $v'$ to path $p$
17:            $pq$.push($newp$, $v'.end$)          ▷ $v'$ covers source block ($start, end$)
Output: backtrace $\#_e.minPath$ will returns the path with minimum distance
```

Figure 3 shows an example of the path-based graph search algorithm. Note that, since the matching between the translation and the references are carried out for each source span, the spurious function word translations will be more carefully examined at this stage, compared previous segment-based approaches, such as BLEU.

4 Experiments

We perform experiments to compare the RGraph-based evaluation metrics we proposed against three standard and popular metrics: 4-gram case-insensitive BLEU, TER and Meteor [12,28,31]. Some statistics of the references before and after the RGraph extension are shown, including the correlation comparisons between the given metrics and human judgments. Then we present tuning comparisons including multiple and single reference cases.

4.1 Correlation with Human Judgments

For the correlation experiments, we use LDC2006T04 (MT03) as the experiment data, which contains 919 source language sentences and 4 references for each source sentence. Additionally, for each source sentence, 6 machine translation outputs are provided, together with their human evaluation scores, ranging from 1 to 5. We use the method in Sect. 2 to build RGraph from the given 4 references for each sentence. Table 1 shows the statistics after the RGraph extension. Besides 67 sentences (7%) which are not extended, more than 90% of the sentences get extra references generated. Among them, 263 sentences (29%) get more than 300 references. The average number of references in the RGraph reaches 125, which cover a significantly larger set of translation candidates compared to the original 4.

Table 1. #sents with different #refs after RGraph extension.

#refs	4	<10	<20	<300	>300	all
#sents	67	168	305	116	263	919
%	7%	18%	33%	13%	29%	100%

Table 2. Comparison of correlations with human judgment.

Conditions	BLEU	TER	Meteor
w. 4-refs	0.4664	0.5066	0.4865
w. RGraph	**0.4739**	**0.5267**	**0.4876**

Table 3. Experiment data and statistics.

Data	Usage	Sentences
LDC	TM train	8,396,924
Gigaword	LM train	14,684,074
MT03	dev	919
MT02	test	878
MT04	test	1,788
MT05	test	1,082

We compute the sentence-level evaluation score for each sentence in the dataset and calculate the correlations between these scores and the human evaluation results (Table 2). The first row shows the evaluation using BLEU, TER and Meteor on the original 4 references; the second row shows the results using the same three metrics but on the RGraph. The correlation efficient improves by a considerable margin for all three metrics, showing that evaluations using RGraph is closer to human judgment.

4.2 Tuning Experiments

To validate the influence of tuning metrics to the whole translation system, we perform tuning experiments on a large-scale machine translation task. Our translation system is an in-house implementation of the hierarchical phrase-based translation system [9], tuned with MERT [26]. The data used to train and test is listed in Table 3. The translation model (TM) of the system is trained on parallel sentences from LDC[1], which consists of 8.3 million of sentence pairs. The Chinese side of the corpora is word segmented using ICTCLAS[2]. We train a 5-gram language model (LM) with MKN smoothing [8], on Xinhua portion of Gigaword. We use multi-reference data MT03 as the development (dev) data, MT02, MT04 and MT05 as the test data. These data are mainly in the same genre, avoiding the extra consideration of domain adaptation. All the reported results are the average of three independent MERT runs with random starting points [10].

We tune the systems with original BLEU and BLEU with the RGraph extension (denoted as GBLEU), respectively, and evaluate the translation result using all previous mentioned metrics (Table 4). Different rows show the evaluation results in different metrics on all three test sets and their average. The left and right half of the table present the system tuned with BLEU and GBLEU, respectively. It could be easily seen that tuning with GBLEU achieves superior performances in all the listed metrics.

[1] including LDC2002E18, LDC2003E14, LDC2004E12, LDC2004T08, LDC2005T10, LDC2007T09.

[2] http://ictclas.nlpir.org/.

Table 4. Comparisons between tuning with BLEU and GBLEU. "ave" denotes the average results across three testsets. "Δ-ave" denotes the difference of scores in each evaluation metric and † indicates statistically significant difference ($p < 0.01$) between systems tuned with GBLEU and BLEU, respectively.

	Tuned with BLEU				Tuned with GBLEU				
Evaluate	MT02	MT04	MT05	ave	MT02	MT04	MT05	ave	Δ-ave
BLEU	37.71	37.28	36.65	37.21	37.77	37.49	36.69	37.31	**+0.10**
1-TER	43.96	43.81	44.07	43.95	44.92†	44.66†	45.04†	44.87	**+0.93**
Meteor	32.88	32.56	33.03	32.82	33.04†	32.70†	33.33†	33.03	**+0.22**
GBLEU	26.90	27.01	28.22	27.38	27.73	27.69	28.95	28.13	**+0.75**
1-GTER	45.57	45.66	45.75	45.66	47.28	47.30	47.36	47.31	**+1.65**
GMeteor	32.83	32.38	32.63	32.61	33.08	32.68	33.06	32.94	**+0.33**

Despite the improvement on RGraph-based metrics, it is interesting to notice that the score increases on all the original metrics computed on the 4 references (by +0.1 in BLEU, +0.93 in TER and +0.22 in Meteor). This improvement could only be explained by the improvement of overall system performance. This result demonstrate that the improvement in evaluation metrics does lead to a stronger statistical system, which may encourage further investigations in the research of evaluation metrics.

4.3 Experiments with a Single Reference

To generate RGraphs from single references, we employ the paraphrase table from Pavlick et al. [29], named PPDB. We use the small size English paraphrase table in PPDB, which has a higher precision, and only use pairs labeled as 'Equivalence' to reduce noisy translations. We construct the monolingual graph D_m using paraphrases for sub-paths that contain less than 3 vertexes, and merge D_m to the D_s of the single reference. Similar with previous experiments, we use BLEU and GBLEU to tune our systems on MT03. The systems are tuned with each reference, and with the monolingual extension of each reference, respectively. The average results are listed in Table 5. Similar with the results in the multi-reference case, the system tuned with GBLEU achieves better results in most of the metrics (by +0.49 in BLEU, +1.40 in TER, etc.). We notice that the results decrease in Meteor. One possible explanation is that the paraphrase matching used by Meteor plays a similar role as our monolingual extension. However, when multiple references is given, our method do explore the references better, even when evaluated with Meteor, as shown in Table 4.

Because systems tuned with different references vary in performance, we further examine those differences (shown in Fig. 4). The error bar shows the maximum and minimum score in the systems tuned by the each single reference. Despite the higher average scores, it is easy to see that systems tuned with GBLEU has a much smaller variance in both BLEU and GBLEU scores,

Table 5. Comparisons between tuning with BLEU and GBLEU on single references. Each score is the results averaged over the system tuned with each of the 4 references. "ave" denotes the average results across the three testsets. "Δ-ave" denotes the difference of scores in each evaluation metric and † indicates statistically significant difference ($p < 0.01$) between systems tuned with GBLEU and BLEU, respectively.

	Tuned with BLEU				Tuned with GBLEU				
Evaluate	MT02	MT04	MT05	ave	MT02	MT04	MT05	ave	Δ-ave
BLEU	36.45	36.36	35.94	36.26	36.94	36.89	36.43	36.75	**+0.49**
1-TER	41.01	41.23	41.54	41.26	42.43†	42.54†	42.99†	42.66	**+1.40**
Meteor	33.34	33.02	33.48	33.28	33.10	32.79	33.25	33.05	−0.23
GBLEU	25.81	25.94	26.97	26.24	26.40	26.47	27.62	26.83	**+0.59**
1-GTER	42.61	43.02	42.80	42.81	44.29	44.53	44.55	44.46	**+1.65**
GMeteor	33.16	32.79	33.00	32.98	32.98	32.63	32.85	32.82	−0.16

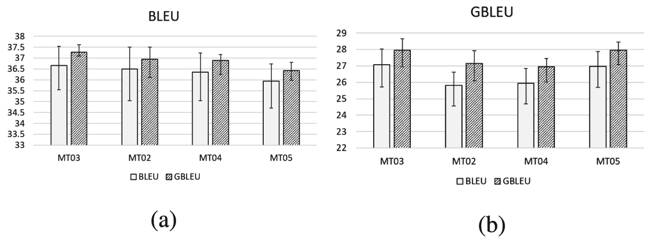


Fig. 4. The BLEU(a) and GBLEU(b) scores of the single reference experiments. The white and shaded pillars indicates system scores (averaged over 4 references) tuned by BLEU and GBLEU, respectively. The error bars depict the minimum and maximum scores tuned on each single reference.

compared to systems tuned with BLEU. This result suggests that the tuning result with single reference may highly depend on the quality of the reference. Using RGraph as the tuning metric helps to reduce this influence and leads to generally more stable results.

5 Related Work

There are several related works focusing on other properties or conditions of the references. Snover et al. [31] proposed to use human edited reference which could achieve evaluation results better correlated with human decisions. Our method tries to automatically generate close references instead. Qin and Specia [30] proposed an approach to explore the information among references. Their work mainly focuses on selecting essential words or expressions using recurring information among references; while our work is to increase the coverage of diverse translations. Albrecht and Hwa [2,3] proposed to use translations from other machine translation systems or large monolingual corpora or tree banks as pseudo references instead of references. They focused on the problem where no human translations are available.

There are other researches proposing tuning metrics for machine translation. Chen et al. [7] proposed a metric named PORT, which combines precision, recall and an ordering metric for better tuning in MT systems. Li et al. [22] used a dependency-based MT evaluation metric RED for Tuning. These methods could also be applied in our RGraph approach. With a closer reference generated by RGraph, it is possible to achieve even better results with these improved tuning methods.

6 Conclusion

This paper aims at properly evaluating the translation quality of machine translation outputs. This is a special problem because different from traditional tasks such as pos-tagging and parsing, there is more than one correct answer in machine translation tasks. It could be an important issue to consider for tasks which have similar properties, such as text summarization, language generation and image caption.

We notice that improving the diversity of the references is important for the evaluation task as well as the tuning of the system. For statistical machine translation systems such as the hierarchical phrase based systems, we have demonstrated that better evaluation metrics do lead to better trained system. It is now interesting to investigate the translation diversity in other architectures such as neural machine translation systems.

We use off-the-shelf alignment tools to obtain the alignment between the references and the source sentence, which is quite noisy for some sentences. We believe that better alignments could lead to even better evaluation performance.

Acknowledgments. This work is supported by the National Science Foundation of China (Nos. 61672277, 61772261), the Jiangsu Research Foundation for Basic Research (No. BK20170074).

References

1. Agarwal, A., Lavie, A.: Meteor, M-BLEU and M-TER: evaluation metrics for high-correlation with human rankings of machine translation output. In: Proceedings of the Third Workshop on Statistical Machine Translation, Columbus, Ohio, pp. 115–118. Association for Computational Linguistics (2008)
2. Albrecht, J., Hwa, R.: Regression for sentence-level MT evaluation with pseudo references. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, pp. 296–303. Association for Computational Linguistics (2007)
3. Albrecht, J., Hwa, R.: The role of pseudo references in MT evaluation. In: Proceedings of the Third Workshop on Statistical Machine Translation, Columbus, Ohio, pp. 187–190. Association for Computational Linguistics, June 2008

4. Banerjee, S., Lavie, A.: METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, Michigan, pp. 65–72. Association for Computational Linguistics (2005)
5. Bangalore, B., Bordel, G., Riccardi, G.: Computing consensus translation from multiple machine translation systems. In: 2001 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2001, pp. 351–354 (2001)
6. Chan, Y.S., Ng, H.T.: MaxSim: performance and effects of translation fluency. Mach. Transl. **23**(2–3), 157–168 (2009)
7. Chen, B., Kuhn, R., Larkin, S.: Port: a precision-order-recall MT evaluation metric for tuning. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Long Papers, vol. 1, pp. 930–939. Association for Computational Linguistics (2012)
8. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. In: Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, California, USA, pp. 310–318. Association for Computational Linguistics (1996)
9. Chiang, D.: A hierarchical phrase-based model for statistical machine translation. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, Ann Arbor, Michigan, pp. 263–270. Association for Computational Linguistics (2005)
10. Clark, J.H., Dyer, C., Lavie, A., Smith, N.A.: Better hypothesis testing for statistical machine translation: controlling for optimizer instability. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, USA, pp. 176–181. Association for Computational Linguistics (2011)
11. Denkowski, M., Lavie, A.: Meteor-next and the meteor paraphrase tables: improved evaluation support for five target languages. In: Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, Uppsala, Sweden, pp. 339–342. Association for Computational Linguistics (2010)
12. Denkowski, M., Lavie, A.: Meteor universal: language specific translation evaluation for any target language. In: Proceedings of the Ninth Workshop on Statistical Machine Translation, Baltimore, Maryland, USA, pp. 376–380. Association for Computational Linguistics (2014)
13. Du, J., Jiang, J., Way, A.: Facilitating translation using source language paraphrase lattices. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, pp. 420–429. Association for Computational Linguistics (2010)
14. Feng, Y., Liu, Y., Mi, H., Liu, Q., Lü, Y.: Lattice-based system combination for statistical machine translation. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, pp. 1105–1113. Association for Computational Linguistics (2009)
15. Han, A.L.F., Wong, D.F., Chao, L.S.: LEPOR: a robust evaluation metric for machine translation with augmented factors, pp. 441–450 (2012)
16. Hopkins, M., May, J.: Tuning as ranking. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK, pp. 1352–1362. Association for Computational Linguistics (2011)
17. Hovy, E.: Toward finely differentiated evaluation metrics for machine translation. In: Proceedings of the EAGLES Workshop on Standards and Evaluation, pp. 127–133 (1999)

18. Jayaraman, S., Lavie, A.: Multi-engine machine translation guided by explicit word matching. In: Proceedings of the ACL Interactive Poster and Demonstration Sessions, Ann Arbor, Michigan, pp. 101–104. Association for Computational Linguistics (2005)
19. Jiang, J., Du, J., Way, A.: Incorporating source-language paraphrases into phrase-based SMT with confusion networks. In: Proceedings of Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation, Portland, Oregon, USA, pp. 31–40. Association for Computational Linguistics (2011)
20. Leusch, G., Ueffing, N., Ney, H.: A novel string-to-string distance measure with applications to machine translation evaluation. In: MT Summit IX, New Orleans, LA, pp. 240–247 (2003)
21. Leusch, G., Ueffing, N., Ney, H.: CDER: efficient MT evaluation using block movements. In: 11th Conference of the European Chapter of the Association for Computational Linguistics (2006)
22. Li, L., Yu, H., Liu, Q.: MT tuning on RED: a dependency-based evaluation metric. In: Proceedings of the Tenth Workshop on Statistical Machine Translation, Lisbon, Portugal, pp. 428–433. Association for Computational Linguistics (2015)
23. Liu, C., Dahlmeier, D., Ng, H.T.: TESLA: translation evaluation of sentences with linear-programming-based analysis. In: Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, Uppsala, Sweden, pp. 354–359. Association for Computational Linguistics (2010)
24. Mariño, J.B., Banchs, R.E., Crego, J.M., de Gispert, A., Lambert, P., Fonollosa, J.A.R., Costa-jussà, M.R.: N-gram-based machine translation. Comput. Linguist. **32**(4), 527–549 (2006)
25. Matusov, E., Ueffing, N., Ney, H.: Computing consensus translation for multiple machine translation systems using enhanced hypothesis alignment. In: 11th Conference of the European Chapter of the Association for Computational Linguistics (2006)
26. Och, F.J.: Minimum error rate training in statistical machine translation. In: Proceedings of ACL 2003, Sapporo, Japan, pp. 160–167. Association for Computational Linguistics (2003)
27. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Comput. Linguist. **29**(1), 19–51 (2003)
28. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of ACL 2002, Philadelphia, Pennsylvania, USA, pp. 311–318. Association for Computational Linguistics (2002)
29. Pavlick, E., Rastogi, P., Ganitkevitch, J., Van Durme, B., Callison-Burch, C.: PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, pp. 425–430. Association for Computational Linguistics (2015). http://www.aclweb.org/anthology/P15-2070
30. Qin, Y., Specia, L.: Truly exploring multiple references for machine translation evaluation. In: 18th Annual Conference of the European Association for Machine Translation, EAMT, Antalya, Turkey (2015)
31. Snover, M., Dorr, B.J., Schwartz, R.: A study of translation edit rate with targeted human annotation. In: Proceedings of AMTA (2006)
32. Snover, M.G., Madnani, N., Dorr, B., Schwartz, R.: TER-Plus: paraphrase, semantic, and alignment enhancements to translation edit rate. Mach. Transl. **23**(2), 117–127 (2009)