RED: A Reference Dependency Based MT Evaluation Metric

Hui Yu[†§] **Xiaofeng Wu**[‡] **Jun Xie**[†] **Wenbin Jiang**[†] **Qun Liu**[‡†] **Shouxun Lin**[†]
[†]Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences
[§]University of Chinese Academy of Sciences
{yuhui,xiejun,jiangwenbin,sxlin}@ict.ac.cn
[‡]CNGL, School of Computing, Dublin City University
{xiaofengwu,qliu}@computing.dcu.ie

Abstract

Most of the widely-used automatic evaluation metrics consider only the local fragments of the references and translations, and they ignore the evaluation on the syntax level. Current syntax-based evaluation metrics try to introduce syntax information but suffer from the poor parsing results of the noisy machine translations. To alleviate this problem, we propose a novel dependency-based evaluation metric which only employs the dependency information of the references. We use two kinds of reference dependency structures: headword chain to capture the long distance dependency information, and fixed and floating structures to capture the local continuous ngram. Experiment results show that our metric achieves higher correlations with human judgments than BLEU, TER and HWCM on WMT 2012 and WMT 2013. By introducing extra linguistic resources and tuning parameters, the new metric gets the state-of-the-art performance which is better than METEOR and SEMPOS on system level, and is comparable with METEOR on sentence level on WMT 2012 and WMT 2013.

1 Introduction

Automatic machine translation (MT) evaluation plays an important role in the evolution of MT. It not only evaluates the performance of MT systems, but also makes the development of MT systems rapider (Och, 2003). According to the type of the employed information, the automatic MT evaluation metrics can be classified into three categories: lexicon-based metrics, syntax-based metrics and semantic-based metrics.

The lexicon-based metrics, such as BLEU (Papineni et al., 2002), TER (Snover et al., 2006), METEOR (Lavie and Agarwal, 2007) and AMBER (Chen and Kuhn, 2011; Chen et al., 2012), are good at capturing the lexicon or phrase level information, e.g. fixed phrases or idioms. But they cannot adequately reflect the syntax similarity. Current efforts in syntax-based metrics, such as the headword chain based metric (HWCM) (Liu and Gildea, 2005), the LFG dependency tree based metric (Owczarzak et al., 2007) and syntactic/semantic-role overlap (Giménez and Màrquez, 2007) , suffer from the parsing of the potentially noisy machine translations, so the improvement of their performance is restricted due to the serious parsing errors. Semantic-based metrics, such as MEANT (Lo et al., 2012; Lo and Wu, 2013), have the similar problem that the accuracy of semantic role labeling (SRL) can also drop due to the errors in translations. To avoid the parsing of potentially noisy translations, the CCG based metric (Mehay and Brew, 2007) only uses the parsing result of reference and employs 2-gram dependents, but it did not achieve the state-of-the-art performance.

In this paper, we propose a novel dependency tree based MT evaluation metric. The new metric only employs the reference dependency tree, leaving the translation unparsed to avoid the error propagation. We use two kinds of reference dependency structures in our metric. One is the headword chain (Liu and Gildea, 2005) which can capture long distance dependency information. The other is fixed and floating structure (Shen et al., 2010) which can capture local continuous ngram. When calculating the matching score between the headword chain and the translation, we use a distance-based similarity. Experiment

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 2042–2051, Dublin, Ireland, August 23-29 2014.

results show that our metric achieves higher correlations with human judgments than BLEU, TER and HWCM on WMT 2012 and WMT 2013. After introducing extra resources and tuning parameters on WMT 2010, the new metric is better than METEOR and SEMPOS on system level and comparable with METEOR on sentence level on WMT 2012 and WMT2013.

The remainder of this paper is organized as follows. Section 2 describes our new reference dependency based MT evaluation metric. In Section 3, we introduce some extra resources to this new metric. Section 4 presents the parameter tuning for the new metric. Section 5 gives the experiment results. Conclusions and future work are discussed in Section 6.

2 RED: A Reference Dependency Based MT Evaluation Metric

The new metric is a REference Dependency based automatic evaluation metric, so we name it RED. We present the new metric detailedly in this section. The description of dependency ngrams is given in Section 2.1. The method to score the dependency ngram is presented in Section 2.2. At last, the method of calculating the final score is introduced in Section 2.3.

2.1 Two Kinds of Dependency Ngrams

To capture both the long distance dependency information and the local continuous ngrams, we use both the headword chain and the fixed-floating structures in our new metric, which correspond to the two kinds of dependency ngram (dep-ngram), headword chain ngram and fixed-floating ngram.

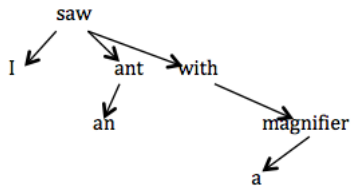

Figure 1: An example of dependency tree.

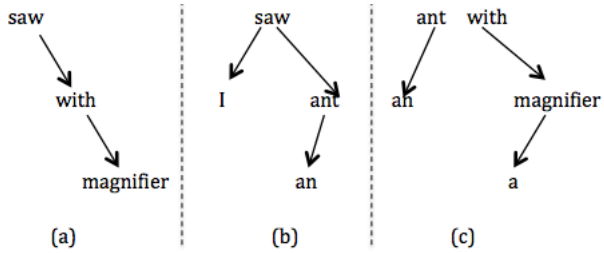

Figure 2: Different kinds of structures extracted from the dependency tree in Figure 1. (a): Headword chain. (b): Fixed structure. (c): Floating structure.

2.1.1 Headword chain

Headword chain is a sequence of words which corresponds to a path in the dependency tree (Liu and Gildea, 2005). For example, Figure 2(a) is a 3-word headword chain extracted from the dependency tree in Figure 1. Headword chain can represent the long distance dependency information, but cannot capture most of the continuous ngrams. In our metric, headword chain corresponds to the headword chain ngram in which the positions of the words are considered. So the form of headword chain ngram is expressed as $(w1_{pos1}, w2_{pos2}, ..., wn_{posn})$, where n is the length of the headword chain ngram. For example, the headword chain in Figure 2(a) is expressed as $(saw_2, with_5, magnifier_7)$.

2.1.2 Fixed and floating structures

Fixed and floating structures are defined in Shen et al. (2010). Fixed structures consist of a sub-root with children, each of which must be a complete constituent. They are called fixed dependency structures because the head is known or fixed. For example, Figure 2(b) shows a fixed structure. Floating structures consist of a number of consecutive sibling nodes of a common head, but the head itself is unspecified. Each of the siblings must be a complete constituent. Figure 2(c) shows a floating structure. Fixed-floating structures correspond to fixed-floating ngrams in our metric. Fixed-floating ngrams don't need the position information, and can be simply expressed as $(w1, w2, ..., wn)$, where n is the length of the

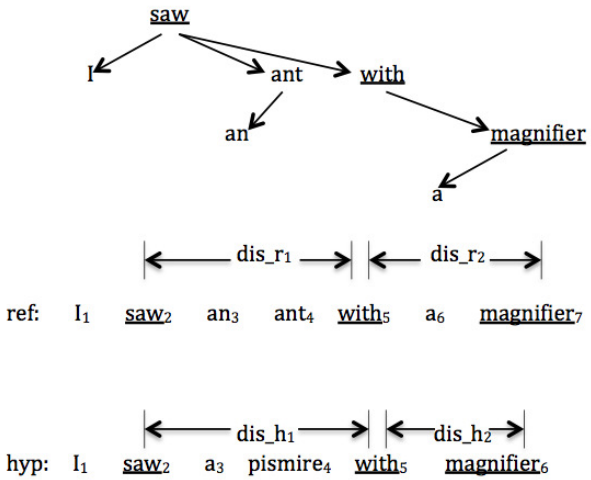

Figure 3: An example of calculating matching score for a headword chain ngram $(saw_2, with_5, magnifier_7)$. dis_r_1 and dis_r_2 are the distances between the corresponding two words in the reference. dis_h_1 and dis_h_2 are the distances between the corresponding two words in the hypothesis.

fixed-floating ngram. For example, the fixed structure in Figure 2(b) and the floating structre in Figure 2(c) can be expressed as (I, saw, an, ant) and $(an, ant, with, a, magnifier)$ respectively.

2.2 Scoring Dep-ngrams

Headword chain ngrams may not be continuous, while fixed-floating ngrams must be continuous. So the scoring methods of the two kinds of dep-ngrams are different, and we introduce the two scoring methods in Section 2.2.1 and Section 2.2.2 respectively.

2.2.1 Scoring headword chain ngram

For a headword chain ngram $(w1_{pos1}, w2_{pos2}, ..., wn_{posn})$, if we can find all these n words in the string of the translation with the same order as they appear in the reference sentence, we consider it a match and the matching score is a distance-based similarity which is calculated by the relative distance, otherwise it is not a match and the score is 0. The matching score is a decimal value between 0 and 1, which is more suitable than just use integer 0 and 1. For example, if the distance between two words in reference is 1, but the distance in two different hypotheses are 2 and 5 respectively. It's more reasonable to score them 0.5 and 0.2 rather than 1 and 0.

The relative distance dis_r_i between every two adjacent words in this kind of dep-ngram is calculated by Formula (1), where pos_{wi} is the position of word wi in the sentence. In Formula (1), we have $1 \leq i \leq n-1$ and n is the length of the dep-ngram. Then a vector $(dis\_r_1, dis\_r_2, ..., dis\_r_{n-1})$ is obtained. In the same way, we obtain vector $(dis\_h_1, dis\_h_2, ..., dis\_h_{n-1})$ for the translation side.

$$dis\_r_i = |pos_{w(i+1)} - pos_{wi}| \tag{1}$$

The matching score $p_{(d,hyp)}$ for a headword chain ngram (*d*) and the translation (*hyp*) is calculated according to Formula (2), where $n > 1$. When the length of the dep-ngram equals 1, the matching score equals 1 if the translation has the same word, otherwise, the matching score equals 0.

$$p_{(d,hyp)} = \begin{cases} exp(-\frac{\sum_{i=1}^{n-1} |dis\_r_i - dis\_h_i|}{n-1}) & if\ match \\ 0 & if\ unmatch \end{cases} \tag{2}$$

An example illustrating the calculation of the matching score $p_{(d,hyp)}$ is shown in Figure 3. There is a 3-word headword chain ngram $(saw_2, with_5, magnifier_7)$ in the dependency tree of the reference.

For this dep-3gram, the words are represented with underline in the reference dependency tree and the reference sentence in Figure 3. We can also find all the same three underlined words in the translation with the same order as they appear in the reference. Therefore, there is a match for this dep-3gram. To compute the matching score between this dep-3gram and the translation, we have:

- Calculate the distance

$$dis\_r_1 = |pos_{with} - pos_{saw}| = |5-2| = 3 \qquad dis\_r_2 = |pos_{magnifier} - pos_{with}| = |7-5| = 2$$
$$dis\_h_1 = |pos_{with} - pos_{saw}| = |5-2| = 3 \qquad dis\_h_2 = |pos_{magnifier} - pos_{with}| = |6-5| = 1$$

- Get the matching score as Formula (3) according to Formula (2). d denotes $(saw_2, with_5, magnifier_7)$ and hyp denotes the translation in the example.

$$p_{(d,hyp)} = exp(-\frac{|dis\_r_1 - dis\_h_1| + |dis\_r_2 - dis\_h_2|}{3-1}) = exp(-\frac{|3-3| + |2-1|}{3-1}) = exp(-0.5) \tag{3}$$

We also tried other methods to calculate the matching score, such as the cosine distance and the absolute distance, but the relative distance performed best. For a headword chain ngram with more than one matches in the translation, we choose the one with the highest matching score.

2.2.2 Scoring fixed-floating ngram

The words in the fixed-floating ngram are continuous, so we restrict the matched string in the translation also to being continuous. That means, for a fixed-floating ngram $(w1, w2, ..., wn)$, if we can find all these n words continuous in the translation with the same order as they appear in the reference, we think the dep-ngram can match with the translation. The matching score can be obtained by Formula (4), where d stands for a fixed-floating ngram and hyp stands for the translation.

$$p_{(d,hyp)} = \begin{cases} 1 & if\ match \\ 0 & if\ unmatch \end{cases} \tag{4}$$

2.3 Scoring RED

In the new metric, we use Fscore to obtain the final score. Fscore is calculated by Formula (5), where α is a value between 0 and 1.

$$Fscore = \frac{precision \cdot recall}{\alpha \cdot precision + (1-\alpha) \cdot recall} \tag{5}$$

The dep-ngrams of the reference and the string of the translation are used to calculate the precision and recall. In order to calculate precision, the number of the dep-ngrams in the translation should be given, but there is no dependency tree for the translation in our method. We know that the number of dep-ngrams has an approximate linear relationship with the length of the sentence, so we use the length of the translation to replace the number of the dep-ngrams in the translation dependency tree. Recall can be calculated directly since we know the number of the dep-ngrams in the reference. The precision and recall are computed as follows.

$$precision = \frac{\sum_{d \in D_n} p_{(d,hyp)}}{len_h}, \quad recall = \frac{\sum_{d \in D_n} p_{(d,hyp)}}{count_{n(ref)}}$$

D_n is the set of dep-ngrams with the length of n. len_h is the length of the translation. $count_{n(ref)}$ is the number of the dep-ngrams with the length of n in the reference.

The final score of RED is achieved using Formula (6), in which a weighted sum of the dep-ngrams' Fscore is calculated. w_{ngram} ($0 \leq w_{ngram} \leq 1$) is the weight of dep-ngram with the length of *n*. Fscore$_n$ is the Fscore for the dep-ngrams with the length of *n*.

$$RED = \sum_{n=1}^{N}(w_{ngram} \times Fscore_n) \tag{6}$$

3 Introducing Extra Resources

Many automatic evaluation metrics can only find the exact match between the reference and the translation, and the information provided by the limited number of references is not sufficient. Some evaluation metrics, such as TERp (Snover et al., 2009) and METOER, introduce extra resources to expand the reference information. We also introduce some extra resources to RED, such as stem, synonym and paraphrase. The words within a sentence can be classified into content words and function words. The effects of the two kinds of words are different and they shouldn't have the same matching score, so we introduce a parameter to distinguish them. The methods of applying these resources are introduced as follows.

- Stem and Synonym

  Stem(Porter, 2001) and synonym (WordNet[1]) are introduced to RED in the following three steps. First, we obtain the alignment with Meteor Aligner (Denkowski and Lavie, 2011) in which not only exact match but also stem and synonym are considered. We use stem and synonym together with exact match as three match modules. Second, the alignment is used to match for a dep-ngram. We think the dep-ngram can match with the translation if the following conditions are satisfied. 1) Each of the words in the dep-ngram has a matched word in the translation according to the alignment; 2) The words in dep-ngram and the matched words in translation appear in the same order; 3) The matched words in translation must be continuous if the dep-ngram is a fixed-floating ngram. At last, the match module score of a dep-ngram is calculated according to Formula (7). Different match modules have different effects, so we give them different weights.

  $$s_{mod} = \frac{\sum_{i=1}^{n} w_{m_i}}{n}, \ \ 0 \leq w_{m_i} \leq 1 \tag{7}$$

  m_i is the match module (exact, stem or synonym) of the ith word in a dep-ngram. w_{m_i} is the match module weight of the ith word in a dep-ngram. n is the number of words in a dep-ngram.

- Paraphrase

  When introducing paraphrase, we don't consider the dependency tree of the reference, because paraphrases may not be contained in the headword chain and fixed-floating structures. First, the alignment is obtained with METEOR Aligner, only considering paraphrase. Second, the matched paraphrases are extracted from the alignment and defined as paraphrase-ngram. The score of a paraphrase is $1 \times w_{par}$, where w_{par} is the weight of paraphrase-ngram.

- Function word

  We introduce a parameter w_{fun} ($0 \leq w_{fun} \leq 1$) to distinguish function words and content words. w_{fun} is the weight of function words. The function word score of a dep-ngram or paraphrase-ngram is computed according to Formula (8).

  $$s_{fun} = \frac{C_{fun} \times w_{fun} + C_{con} \times (1 - w_{fun})}{C_{fun} + C_{con}} \tag{8}$$

  C_{fun} is the number of function words in the dep-ngram or paraphrase-ngram. C_{con} is the number of content words in the dep-ngram or paraphrase-ngram.

[1] http://wordnet.princeton.edu/

We use RED-plus (REDp) to represent RED with extra resources, and the final score are calculated as Formula (9), in which $Fscore_p$ is obtained using $precison_p$ and $recall_p$ as Formula (10).

$$REDp = \sum_{n=1}^{N} (w_{ngram} \times Fscore_{p_n}) \tag{9}$$

$$Fscore_p = \frac{precision_p \cdot recall_p}{\alpha \cdot precision_p + (1-\alpha) \cdot recall_p} \tag{10}$$

$precision_p$ and $recall_P$ in Formula (10) are calculated as follows.

$$precision_p = \frac{score_{par_n} + score_{dep_n}}{len_h}\ , \quad recall_p = \frac{score_{par_n} + score_{dep_n}}{count_n(ref) + count_n(par)}$$

len_h is the length of the translation. $count_{n(ref)}$ is the number of the dep-ngrams with the length of n in the reference. $count_n(par)$ is the number of paraphrases with length of n in reference. $score_{par_n}$ is the match score of paraphrase-ngrams with the length of *n*. $score_{dep_n}$ is the match score of dep-ngrams with the length of *n*. $score_{par_n}$ and $score_{dep_n}$ are calculated as follows.

$$score_{par_n} = \sum_{par \in P_n} (1 \times w_{par} \times s_{fum})\ , \qquad score_{dep_n} = \sum_{d \in D_n} (p_{(d,hyp)} \times s_{mod} \times s_{fun})$$

P_n is the set of paraphrase-ngrams with the length of *n*. D_n is the set of dep-ngrams with the length of *n*.

4 Parameter Tuning

There are several parameters in REDp, and different parameter values can make the performance of REDp different. For example,w_{ngram} represents the weight of dep-ngram with the length of *n*. The effect of ngrams with different lengths are different, and they shouldn't have the same weight. So we can tune the parameters to find their best values.

We try a preliminary optimization method to tune parameters in REDp. A heuristic search is employed and the parameters are classified into two subsets. The parameter optimization is a grid search over the two subsets of parameters. When searching Subset 1, the parameters in Subset 2 are fixed, and then Subset 1 and Subset 2 are exchanged to finish this iteration. Several iterations are executed to finish the parameter tuning process. This heuristic search may not find the global optimum but it can save a lot of time compared with exhaustive search. The optimization goal is to maximize the sum of Spearman's ρ rank correlation coefficient on system level and Kendall's τ correlation coefficient on sentence level. ρ is calculated using the following equation.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2-1)}$$

where d_i is the difference between the human rank and metric's rank for system i. n is the number of systems. τ is calculated as follows.

$$\tau = \frac{\text{number of concordant pairs} - \text{number of discordant pairs}}{\text{number of concordant pairs} + \text{number of discordant pairs}}$$

The data of into-English tasks in WMT 2010 are used to tune parameters. The tuned parameters are listed in Table 1.

5 Experiments

5.1 Data

The test sets in experiments are WMT 2012 and WMT 2013. The language pairs are German-to-English (de-en), Czech-to-English (cz-en), French-to-English (fr-en), Spanish-to-English (es-en) and Russian-to-English (ru-en). The number of translation systems for each language pair are showed in Table 2. For each language pair, there are 3003 sentences in WMT 2012 and 3000 sentences in WMT 2013.

Parameter	α	w_{fun}	w_{exact}	w_{stem}	w_{syn}	w_{par}	w_{1gram}	w_{2gram}	w_{3gram}
tuned values	0.9	0.2	0.9	0.6	0.6	0.6	0.6	0.5	0.1

Table 1: Parameter values after tuning on WMT 2010. α is from Formula (10). w_{fun} is the weight of function word. w_{exact}, w_{stem} and w_{syn} are the weights of the three match modules 'exact stem synonym' respectively. w_{par} is the weight of paraphrase-ngram. w_{1gram}, w_{2gram} and w_{3gram} are the weights of dep-ngram with the length of 1, 2 and 3 respectively.

Language pairs	cz-en	de-en	es-en	fr-en	ru-en
WMT2012	6	16	12	15	-
WMT2013	12	23	17	19	23

Table 2: The number of translation systems for each language pair on WMT 2012 and WMT 2013.

We parsed the reference into constituent tree by Berkeley parser[2] and then converted the constituent tree into dependency tree by Penn2Malt[3]. Presumably, the performance of the new metric will be better if the dependency trees are labeled by human. Reference dependency trees are labeled only once and can be used forever so it will not increase costs.

5.2 Baselines

In the experiments, we compare the performance of our metric with the widely-used lexicon-based metrics such as BLEU[4], TER[5] and METEOR[6], dependency-based metric HWCM and semantic-based metric SEMPOS (Macháček and Bojar, 2011) which has the best performance on system level according to the published results of WMT 2012.

The results of BLEU are obtained using 4-gram with smoothing option. The version of TER is 0.7.25. The results of METEOR are obtained by Version 1.4 with task option '*rank*'. We re-implement HWCM which employs an epsilon value of 10^{-3} to replace zero for smoothing purpose. The correlations of SEMPOS are obtained from the published results of WMT 2012 and WMT 2013.

5.3 Experiment Results

The experiments on both system level and sentence level are carried out. On system level, the correlations are calculated using Spearman's rank correlation coefficient ρ (Pirie, 1988). Kendall's rank correlation coefficient τ (Kendall, 1938) is employed to evaluate the sentence level correlation. Our method performs best when the maximum length of dep-ngram is set to 3, so we only present the results with the maximum length of 3. RED represents the new metric with exact match and the parameter values are set as follows. $\alpha = 0.5$. $w_{1gram} = w_{2gram} = w_{3gram} = 1/3$. REDp represents the new metric with extra resources and tuned parameter values which are listed in Table (1).

5.3.1 System level correlations

The system level correlations are shown in Table 3. RED is better than BLEU, TER and HWCM on average on both WMT 2012 and WMT 2013, which reflects that using syntactic information and only parsing the reference side are helpful. REDp gets the best result on all of the language pairs except cz-en on WMT 2012. The significant improvement from RED to REDp illustrates the effect of extra resources and the parameter tuning. Stem, synonym and paraphrase can enrich the reference and provide extra knowledge for automatic evaluation metric. There are several parameters in REDp, and different parameter values can make the performance of REDp different. So the performance can be optimized through parameter tuning. SEMPOS got the best correlation according to the published results of WMT

[2] http://code.google.com/p/berkeleyparser/downloads/list
[3] http://stp.lingfil.uu.se/~nivre/research/Penn2Malt.html
[4] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a.pl
[5] http://www.cs.umd.edu/~snover/tercom
[6] http://www.cs.cmu.edu/~alavie/METEOR/download/meteor-1.4.tgz

2012, and METEOR got the best correlation according to the published results of WMT 2013 on into-English task on system level. REDp gets better result than SEMPOS and METEOR on both WMT 2012 and WMT 2013, so REDp achieves the state-of-the-art performance on system level.

data	WMT 2012					WMT 2013					
Metrics	cz-en	de-en	es-en	fr-en	ave	cz-en	de-en	es-en	fr-en	ru-en	ave
BLEU	.886	.671	.874	.811	.811	.936	.895	.888	.989	.670	.876
TER	.886	.624	.916	.821	.812	.800	.833	.825	.951	.581	.798
HWCM	.943	.762	.937	.818	.865	.902	.904	.886	.951	.756	.880
METEOR	.657	.885	.951	**.843**	.834	.964	.961	.979	.984	.789	.935
SEMPOS	.943	.924	.937	.804	.902	.955	.919	.930	.938	.823	.913
RED	**1.0**	.759	.951	.818	.882	.964	.951	.930	.989	.725	.912
REDp	.943	**.947**	**.965**	**.843**	**.925**	**.982**	**.973**	**.986**	**.995**	**.800**	**.947**

Table 3: System level correlations on WMT 2012 and WMT 2013. The value in bold is the best result in each column. *ave* stands for the average result of the language pairs on WMT 2012 or WMT 2013.

5.3.2 Sentence level correlations

The sentence level correlations on WMT 2012 and WMT 2013 are shown in Table 4. RED is better than BLEU and HWCM on all the language pairs, which reflects the effectiveness of syntactic information and only parsing the reference. By introducing extra resources and parameter tuning, REDp achieves significant improvement over RED. Stem, synonym and paraphrase can enrich the reference and provide extra knowledge for automatic evaluation metric. There are several parameters in REDp, and different parameter values can make the performance of REDp different. A better performance can be exploited through parameter tuning. From the results of REDp and METEOR, we can see that REDp gets the comparable results with METEOR on sentence level on both WMT 2012 and WMT 2013.

data	WMT 2012					WMT 2013					
Metrics	cz-en	de-en	es-en	fr-en	ave	cz-en	de-en	es-en	fr-en	ru-en	ave
BLEU	.157	.191	.189	.210	.187	.199	.220	.259	.224	.162	.213
HWCM	.158	.207	.203	.204	.193	.187	.208	.247	.227	.175	.209
METEOR	**.212**	**.275**	**.249**	**.251**	**.247**	**.265**	**.293**	**.324**	**.264**	**.239**	**.277**
RED	.165	.218	.203	.221	.202	.210	.239	.292	.246	.196	.237
REDp	**.212**	.271	.234	.250	.242	.259	.290	.323	.260	.223	.271

Table 4: Sentence level correlations on WMT 2012 and WMT 2013. The value in bold is the best result in each column. *ave* stands for the average result of the language pairs on WMT 2012 or WMT 2013.

6 Conclusion and Future Work

In this paper, we propose a reference dependency based automatic MT evaluation metric RED. The new metric only uses the dependency trees of the reference, which avoids the parsing of the potentially noisy translations. Both long distance dependency information and the local continuous ngrams are captured by the new metric. The experiment results indicate that RED achieves better correlations than BLEU, TER and HWCM on both system level and sentence level. REDp, the improved version of RED through adding extra resources and preliminary parameter tuning, gets state-of-the-art results which are better than METEOR and SEMPOS on system level. On sentence level, REDp gets the comparable performance with METEOR.

In the future, we will use the dependency forest instead of the dependency tree to reduce the effect of parsing errors. We will also apply RED and REDp to the tuning process of SMT to improve the translation quality.

Acknowledgements

The authors were supported by National Natural Science Foundation of China (Contract 61202216) and National Natural Science Foundation of China (Contract 61379086). 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. Sincere thanks to the three anonymous reviewers for their thorough reviewing and valuable suggestions.

References

Boxing Chen and Roland Kuhn. 2011. Amber: A modified bleu, enhanced ranking metric. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 71–77, Edinburgh, Scotland, July. Association for Computational Linguistics.

Boxing Chen, Roland Kuhn, and George Foster. 2012. Improving amber, an mt evaluation metric. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, WMT '12, pages 59–63, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91. Association for Computational Linguistics.

Jesús Giménez and Lluís Màrquez. 2007. Linguistic features for automatic evaluation of heterogenous mt systems. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 256–264. Association for Computational Linguistics.

Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: an automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 228–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

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 and/or Summarization*, pages 25–32.

Chi-kiu Lo and Dekai Wu. 2013. MEANT at WMT 2013: A tunable, accurate yet inexpensive semantic frame based MT evaluation metric. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 422–428, Sofia, Bulgaria, August. Association for Computational Linguistics.

Chi-kiu Lo, Anand Karthik Tumuluru, and Dekai Wu. 2012. Fully automatic semantic mt evaluation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 243–252, Montréal, Canada, June. Association for Computational Linguistics.

Matouš Macháček and Ondřej Bojar. 2011. Approximating a deep-syntactic metric for mt evaluation and tuning. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 92–98. Association for Computational Linguistics.

Dennis Mehay and Chris Brew. 2007. BLEUTRE: Flattening Syntactic Dependencies for MT Evaluation. In *Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*.

F.J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Karolina Owczarzak, Josef van Genabith, and Andy Way. 2007. Dependency-based automatic evaluation for machine translation. In *Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation*, SSST '07, pages 80–87, Stroudsburg, PA, USA. Association for Computational Linguistics.

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

W Pirie. 1988. Spearman rank correlation coefficient. *Encyclopedia of statistical sciences*.

Martin F Porter. 2001. Snowball: A language for stemming algorithms.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2010. String-to-dependency statistical machine translation. *Computational Linguistics*, 36(4):649–671.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Matthew Snover, Nitin Madnani, Bonnie J Dorr, and Richard Schwartz. 2009. Fluency, adequacy, or hter?: exploring different human judgments with a tunable mt metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 259–268. Association for Computational Linguistics.