Stochastic Iterative Alignment for Machine Translation Evaluation

Ding Liu and **Daniel Gildea**
Department of Computer Science
University of Rochester
Rochester, NY 14627

Abstract

A number of metrics for automatic evaluation of machine translation have been proposed in recent years, with some metrics focusing on measuring the adequacy of MT output, and other metrics focusing on fluency. Adequacy-oriented metrics such as BLEU measure n-gram overlap of MT outputs and their references, but do not represent sentence-level information. In contrast, fluency-oriented metrics such as ROUGE-W compute longest common subsequences, but ignore words not aligned by the LCS. We propose a metric based on stochastic iterative string alignment (SIA), which aims to combine the strengths of both approaches. We compare SIA with existing metrics, and find that it outperforms them in overall evaluation, and works specially well in fluency evaluation.

1 Introduction

Evaluation has long been a stumbling block in the development of machine translation systems, due to the simple fact that there are many correct translations for a given sentence. Human evaluation of system output is costly in both time and money, leading to the rise of automatic evaluation metrics in recent years. In the 2003 Johns Hopkins Workshop on Speech and Language Engineering, experiments on MT evaluation showed that BLEU and NIST do not correlate well with human judgments at the sentence level, even when they correlate well over large test sets (Blatz et al., 2003). Liu and Gildea (2005) also pointed out that due to the limited references for every MT output, using the overlapping ratio of n-grams longer than 2 did not improve sentence level evaluation performance of BLEU. The problem leads to an even worse result in BLEU'S fluency evaluation, which is supposed to rely on the long n-grams. In order to improve sentence-level evaluation performance, several metrics have been proposed, including ROUGE-W, ROUGE-S (Lin and Och, 2004) and METEOR (Banerjee and Lavie, 2005). ROUGE-W differs from BLEU and NIST in that it doesn't require the common sequence between MT output and the references to be consecutive, and thus longer common sequences can be found. There is a problem with loose-sequence-based metrics: the words outside the longest common sequence are be considered in the metric, even if they appear both in MT output and the reference. ROUGE-S is meant to alleviate this problem by computing the common skipped bigrams instead of the LCS. But the price ROUGE-S pays is falling back to the shorter sequences and losing the advantage of long common sequences. METEOR is essentially a unigram based metric, which prefers the monotonic word alignment between MT output and the references by penalizing crossing word alignments. There are two problems with METEOR. First, it doesn't consider gaps in the aligned words, which is an important feature for evaluating the sentence fluency; second, it cannot use multiple references simultaneously.[1] ROUGE and METEOR both use WordNet and Porter Stemmer to increase the chance of the MT output words matching the reference words. Such morphological processing and synonym extraction tools are available for English, but are not always available for other languages. In order to take advantage of loose-sequence-based metrics and avoid the problems in ROUGE and METEOR, we propose a new metric SIA, which is based on loose sequence alignment but enhanced with the following features:

[1] METEOR and ROUGE both compute the score based on the best reference

- Computing the string alignment score based on the gaps in the common sequence. Though ROUGE-W also takes into consider the gaps in the common sequence between the MT output and the reference by giving more credits to the n-grams in the common sequence, our method is more flexible in that not only do the strict n-grams get more credits, but also the tighter sequences.

- Stochastic word matching. For the purpose of increasing hitting chance of MT outputs in references, we use a stochastic word matching in the string alignment instead of WORD-STEM and WORD-NET used in METEOR and ROUGE. Instead of using exact matching, we use a soft matching based on the similarity between two words, which is trained in a bilingual corpus. The corpus is aligned in the word level using IBM Model4 (Brown et al., 1993). Stochastic word matching is a uniform replacement for both morphological processing and synonym matching. More importantly, it can be easily adapted for different kinds of languages, as long as there are bilingual parallel corpora available (which is always true for statistical machine translation).

- Iterative alignment scheme. In this scheme, the string alignment will be continued until there are no more co-occuring words to be found between the MT output and any one of the references. In this way, every co-occuring word between the MT output and the references can be considered and contribute to the final score, and multiple references can be used simultaneously.

The remainder of the paper is organized as follows: section 2 gives a recap of BLEU, ROUGE-W and METEOR; section 3 describes the three components of SIA; section 4 compares the performance of different metrics based on experimental results; section 5 presents our conclusion.

2 Recap of BLEU, ROUGE-W and METEOR

The most commonly used automatic evaluation metrics, BLEU (Papineni et al., 2002) and NIST (Doddington, 2002), are based on the assumption that "The closer a machine translation is to a professional human translation, the better it is" (Papineni et al., 2002). For every hypothesis, BLEU computes the fraction of n-grams which also appear in the reference sentences, as well as a brevity penalty. NIST uses a similar strategy to BLEU but further considers that n-grams with different frequency should be treated differently in the evaluation (Doddington, 2002). BLEU and NIST have been shown to correlate closely with human judgments in ranking MT systems with different qualities (Papineni et al., 2002; Doddington, 2002).

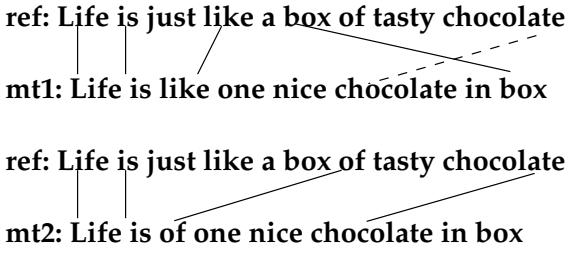

Figure 1: Alignment Example for ROUGE-W

ROUGE-W is based on the weighted longest common subsequence (LCS) between the MT output and the reference. The common subsequences in ROUGE-W are not necessarily strict n-grams, and gaps are allowed in both the MT output and the reference. Because of the flexibility, long common subsequences are feasible in ROUGE-W and can help to reflect the sentence-wide similarity of MT output and references. ROUGE-W uses a weighting strategy where the LCS containing strict n-grams is favored. Figure 1 gives two examples that show how ROUGE-W searches for the LCS. For *mt1*, ROUGE-W will choose either *life is like chocolate* or *life is like box* as the LCS, since neither of the sequences 'like box' and 'like chocolate' are strict n-grams and thus make no difference in ROUGE-W (the only strict n-grams in the two candidate LCS is *life is*). For *mt2*, there is only one choice of the LCS: *life is of chocolate*. The LCS of *mt1* and *mt2* have the same length and the same number of strict n-grams, thus they get the same score in ROUGE-W. But it is clear to us that *mt1* is better than *mt2*. It is easy to verify that *mt1* and *mt2* have the same number of common 1-grams, 2-grams, and skipped 2-grams with the reference (they don't have common n-grams longer than 2 words), thus BLEU and ROUGE-S are also not able to differentiate them.

METEOR is a metric sitting in the middle of the n-gram based metrics and the loose se-

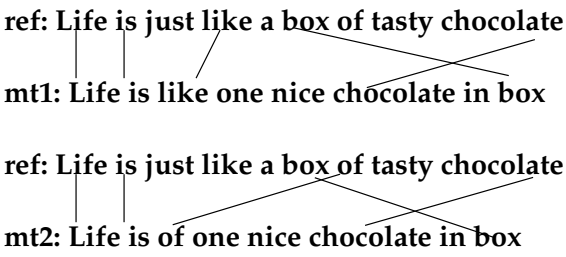

Figure 2: Alignment Example for METEOR

quence based metrics. It has several phases and in each phase different matching techniques (EXACT, PORTER-STEM, WORD-NET) are used to make an alignment for the MT output and the reference. METEOR doesn't require the alignment to be monotonic, which means crossing word mappings (e.g. a b is mapped to b a) are allowed, though doing so will get a penalty. Figure 2 shows the alignments of METEOR based on the same example as ROUGE. Though the two alignments have the same number of word mappings, *mt2* gets more crossed word mappings than *mt1*, thus it will get less credits in METEOR. Both ROUGE and METEOR normalize their evaluation result based on the MT output length (precision) and the reference length (recall), and the final score is computed as the F-mean of them.

3 Stochastic Iterative Alignment (SIA) for Machine Translation Evaluation

We introduce three techniques to allow more sensitive scores to be computed.

3.1 Modified String Alignment

This section introduces how to compute the string alignment based on the word gaps. Given a pair of strings, the task of string alignment is to obtain the longest monotonic common sequence (where gaps are allowed). SIA uses a different weighting strategy from ROUGE-W, which is more flexible. In SIA, the alignments are evaluated based on the geometric mean of the gaps in the reference side and the MT output side. Thus in the dynamic programming, the state not only includes the current covering length of the MT output and the reference, but also includes the last aligned positions in them. The algorithm for computing the alignment score in SIA is described in Figure 3. The subroutine COMPUTE_SCORE, which computes the score gained from the current aligned positions, is shown in Figure 4. From the algorithm, we can

function GET_ALIGN_SCORE(*mt*, *M*, *ref*, *N*)
▷ Compute the alignment score of the MT output *mt* with length *M* and the reference *ref* with length *N*
for $i = 1; i \leq M; i = i + 1$ **do**
for $j = 1; j \leq N; j = j + 1$ **do**
for $k = 1; k \leq i; k = k + 1$ **do**
for $m = 1; m \leq j; m = m + 1$ **do**
$score_{i,j,k,m} = \max\{score_{i-1,j,k,m}, score_{i,j-1,k,m}\}$;
end for
end for
$score_{i,j,i,j} = \max_{n=1,M;p=1,N}\{score_{i,j,i,j}, score_{i-1,j-1,n,p} + \text{COMPUTE\_SCORE}(mt, ref, i, j, n, p)\}$;
end for
end for
return $\frac{score_{M,N,M,N}}{M}$;
end function

Figure 3: Alignment Algorithm Based on Gaps

function COMPUTE_SCORE(*mt*, *ref*, *i*, *j*, *n*, *p*)
if $mt[i] == ref[j]$ **then**
return $1/\sqrt{(i-n)\times(j-p)}$;
else
return 0;
end if
end function

Figure 4: Compute Word Matching Score Based on Gaps

see that not only will strict n-grams get higher scores than non-consecutive sequences, but also the non-consecutive sequences with smaller gaps will get higher scores than those with larger gaps. This weighting method can help SIA capture more subtle difference of MT outputs than ROUGE-W does. For example, if SIA is used to align *mt1* and *ref* in Figure 1, it will choose *life is like box* instead of *life is like chocolate*, because the average distance of 'box-box' to its previous mapping 'like-like' is less than 'chocolate-chocolate'. Then the score SIA assigns to *mt1* is:

$$\left(\frac{1}{1\times 1}+\frac{1}{1\times 1}+\frac{1}{\sqrt{1\times 2}}+\frac{1}{\sqrt{2\times 5}}\right)\times\frac{1}{8}=0.399 \quad (1)$$

For *mt2*, there is only one possible alignment, its score in SIA is computed as:

$$\left(\frac{1}{1\times 1}+\frac{1}{1\times 1}+\frac{1}{\sqrt{1\times 5}}+\frac{1}{\sqrt{2\times 3}}\right)\times\frac{1}{8}=0.357 \quad (2)$$

Thus, *mt1* will be considered better than *mt2* in SIA, which is reasonable. As mentioned in section 1, though loose-sequence-based metrics give a better reflection of the sentence-wide similarity of the MT output and the reference, they cannot

make full use of word-level information. This defect could potentially lead to a poor performance in adequacy evaluation, considering the case that the ignored words are crucial to the evaluation. In the later part of this section, we will describe an iterative alignment scheme which is meant to compensate for this defect.

3.2 Stochastic Word Mapping

In ROUGE and METEOR, PORTER-STEM and WORD-NET are used to increase the chance of the MT output words matching the references. We use a different stochastic approach in SIA to achieve the same purpose. The string alignment has a good dynamic framework which allows the stochastic word matching to be easily incorporated into it. The stochastic string alignment can be implemented by simply replacing the function COMPUTE_SCORE with the function of Figure 5. The function *similarity(word1, word2)* returns a ratio which reflects how similar the two words are. Now we consider how to compute the similarity ratio of two words. Our method is motivated by the phrase extraction method of Bannard and Callison-Burch (2005), which computes the similarity ratio of two words by looking at their relationship with words in another language. Given a bilingual parallel corpus with aligned sentences, say English and French, the probability of an English word given a French word can be computed by training word alignment models such as IBM Model4. Then for every English word *e*, we have a set of conditional probabilities given each French word: $p(e|f_1)$, $p(e|f_2)$, ... , $p(e|f_N)$. If we consider these probabilities as a vector, the similarities of two English words can be obtained by computing the dot product of their corresponding vectors.[2] The formula is described below:

$$similarity(e_i, e_j) = \sum_{k=1}^{N} p(e_i|f_k)p(e_j|f_k) \quad (3)$$

Paraphrasing methods based on monolingual parallel corpora such as (Pang et al., 2003; Barzilay and Lee, 2003) can also be used to compute the similarity ratio of two words, but they don't have as rich training resources as the bilingual methods do.

[2]Although the marginalized probability (over all French words) of an English word given the other English word ($\sum_{k=1}^{N} p(e_i|f_k)p(f_k|e_j)$) is a more intuitive way of measuring the similarity, the dot product of the vectors $p(e|f)$ described above performed slightly better in our experiments.

```
function STO_COMPUTE_SCORE(mt, ref, i, j, n, p)
    if mt[i] == ref[j] then
        return 1/sqrt((i − n) × (j − p));
    else
        return similarity(mt[i],ref[i]) / sqrt((i−n)×(j−p));
    end if
end function
```

Figure 5: Compute Stochastic Word Matching Score

3.3 Iterative Alignment Scheme

ROUGE-W, METEOR, and WER all score MT output by first computing a score based on each available reference, and then taking the highest score as the final score for the MT output. This scheme has the problem of not being able to use multiple references simultaneously. The iterative alignment scheme proposed here is meant to alleviate this problem, by doing alignment between the MT output and one of the available references until no more words in the MT output can be found in the references. In each alignment round, the score based on each reference is computed and the highest one is taken as the score for the round. Then the words which have been aligned in best alignment will not be considered in the next round. With the same number of aligned words, the MT output with fewer alignment rounds should be considered better than those requiring more rounds. For this reason, a decay factor α is multiplied with the scores of each round. The final score of the MT output is then computed by summing the weighted scores of each alignment round. The scheme is described in Figure 6.

The function GET_ALIGN_SCORE_1 used in GET_ALIGN_SCORE_IN_MULTIPLE_REFS is slightly different from GET_ALIGN_SCORE described in the prior subsection. The dynamic programming algorithm for getting the best alignment is the same, except that it has two more tables as input, which record the unavailable positions in the MT output and the reference. These positions have already been used in the prior best alignments and should not be considered in the ongoing alignment. It also returns the aligned positions of the best alignment. The pseudocode for GET_ALIGN_SCORE_1 is shown in Figure 7. The computation of the length penalty is similar to BLEU: it is set to 1 if length of the MT output is longer than the arithmetic mean of length of the

function GET_ALIGN_SCORE_IN_MULTIPLE_REFS(*mt*, ref_1, ..., ref_N, α)
▷ Iteratively Compute the Alignment Score Based on Multiple References and the Decay Factor α
final_score = 0;
while *max_score* != 0 **do**
for *i* = 1, ..., *N* **do**
(*score*, *align*) = GET_ALIGN_SCORE_1(*mt*, ref_i, *mt_table*, ref_table_i);
if *score* > *max_score* **then**
max_score = *score*;
max_align = *align*;
max_ref = *i*;
end if
end for
final_score += *max_score* $\times\alpha$;
$\alpha \times = \alpha$;
Add the words in *align* to *mt_table* and $ref\_table_{max\_ref}$;
end while
return *final_score*$\times$ *length_penalty*;
end function

Figure 6: Iterative Alignment Scheme

function GET_ALIGN_SCORE1(*mt*, *ref*, *mttable*, *reftable*)
▷ Compute the alignment score of the MT output *mt* with length *M* and the reference *ref* with length *N*, without considering the positions in *mttable* and *reftable*
$M = |mt|$; $N = |ref|$;
for $i = 1; i \leq M; i = i+1$ **do**
for $j = 1; j \leq N; j = j+1$ **do**
for $k = 1; k \leq i; k = k+1$ **do**
for $m = 1; m \leq j; m = m+1$ **do**
$score_{i,j,k,m} = \max\{score_{i-1,j,k,m}, score_{i,j-1,k,m}\}$;
end for
end for
if *i* is not in *mttable* and *j* is not in *reftable* **then**
$score_{i,j,i,j} = \max_{n=1,M;p=1,N}\{score_{i,j,i,j}, score_{i-1,j-1,n,p} + \text{COMPUTE\_SCORE}(mt, ref, i, j, n, p)\}$;
end if
end for
end for
return $\frac{score_{M,N,M,N}}{M}$ and the corresponding alignment;
end function

Figure 7: Alignment Algorithm Based on Gaps Without Considering Aligned Positions

references, and otherwise is set to the ratio of the two. Figure 8 shows how the iterative alignment scheme works with an evaluation set containing one MT output and two references. The selected alignment in each round is shown, as well as the unavailable positions in MT output and references. With the iterative scheme, every common word between the MT output and the reference set can make a contribution to the metric, and by such means SIA is able to make full use of the word-level information. Furthermore, the order (alignment round) in which the words are aligned provides a way to weight them. In BLEU, multiple references can be used simultaneously, but the common n-grams are treated equally.

4 Experiments

Evaluation experiments were conducted to compare the performance of different metrics including BLEU, ROUGE, METEOR and SIA.[3] The test data for the experiments are from the MT evaluation workshop at ACL05. There are seven sets of MT outputs (E09 E11 E12 E14 E15 E17 E22), all of which contain 919 English sentences. These sentences are the translation of the same Chinese input generated by seven different MT systems. The fluency and adequacy of each sentence are manually ranked from 1 to 5. For each MT output, there are two sets of human scores available, and

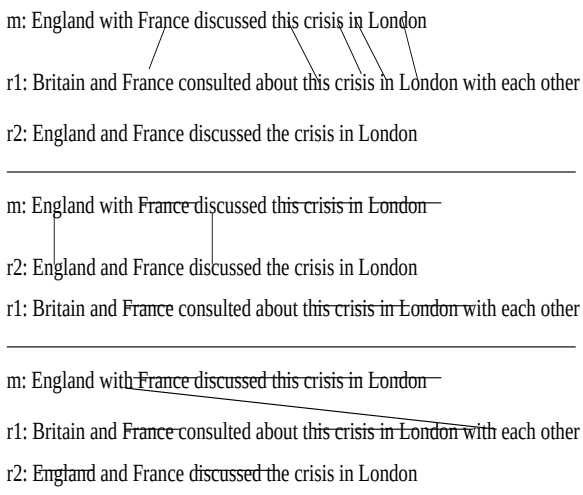

Figure 8: Alignment Example for SIA

[3]METEOR and ROUGE can be downloaded at http://www.cs.cmu.edu/~alavie/METEOR and http://www.isi.edu/licensed-sw/see/rouge

we randomly choose one as the score used in the experiments. The human overall scores are calculated as the arithmetic means of the human fluency scores and adequacy scores. There are four sets of human translations (E01, E02, E03, E04) serving as references for those MT outputs. The MT outputs and reference sentences are transformed to lower case. Our experiments are carried out as follows: automatic metrics are used to evaluate the MT outputs based on the four sets of references, and the Pearson's correlation coefficient of the automatic scores and the human scores is computed to see how well they agree.

4.1 N-gram vs. Loose Sequence

One of the problems addressed in this paper is the different performance of n-gram based metrics and loose-sequence-based metrics in sentence-level evaluation. To see how they really differ in experiments, we choose BLEU and ROUGE-W as the representative metrics for the two types, and used them to evaluate the 6433 sentences in the 7 MT outputs. The Pearson correlation coefficients are then computed based on the 6433 samples. The experimental results are shown in Table 1. BLEU-n denotes the BLEU metric with the longest n-gram of length n. F denotes fluency, A denotes adequacy, and O denotes overall. We see that with the increase of n-gram length, BLEU's performance does not increase monotonically. The best result in adequacy evaluation is achieved at 2-gram and the best result in fluency is achieved at 4-gram. Using n-grams longer than 2 doesn't buy much improvement for BLEU in fluency evaluation, and does not compensate for the loss in adequacy evaluation. This confirms Liu and Gildea (2005)'s finding that in sentence level evaluation, long n-grams in BLEU are not beneficial. The loose-sequence-based ROUGE-W does much better than BLEU in fluency evaluation, but it does poorly in adequacy evaluation and doesn't achieve a significant improvement in overall evaluation. We speculate that the reason is that ROUGE-W doesn't make full use of the available word-level information.

4.2 METEOR vs. SIA

SIA is designed to take the advantage of loose-sequence-based metrics without losing word-level information. To see how well it works, we choose E09 as the development set and the sentences in the other 6 sets as the test data. The decay factor in SIA is determined by optimizing the overall evaluation for E09, and then used with SIA to evaluate the other 5514 sentences based on the four sets of references. The similarity of English words is computed by training IBM Model 4 in an English-French parallel corpus which contains seven hundred thousand sentence pairs. For every English word, only the entries of the top 100 most similar English words are kept and the similarity ratios of them are then re-normalized. The words outside the training corpus will be considered as only having itself as its similar word. To compare the performance of SIA with BLEU, ROUGE and METEOR, the evaluation results based on the same testing data is given in Table 2. B-3 denotes BLEU-3; R_1 denotes the skipped bigram based ROUGE metric which considers all skip distances and uses PORTER-STEM; R_2 denotes ROUGE-W with PORTER-STEM; M denotes the METEOR metric using PORTER-STEM and WORD-NET synonym; S denotes SIA.

	B-3	R_1	R_2	M	S
F	0.167	0.152	0.192	0.167	0.202
A	0.306	0.304	0.287	0.332	0.322
O	0.265	0.256	0.266	0.280	0.292

Table 2: Sentence level evaluation results of BLEU, ROUGE, METEOR and SIA

We see that METEOR, as the other metric sitting in the middle of n-gram based metrics and loose sequence metrics, achieves improvement over BLEU in both adequacy and fluency evaluation. Though METEOR gets the best results in adequacy evaluation, in fluency evaluation, it is worse than the loose-sequence-based metric ROUGE-W-STEM. SIA is the only one among the 5 metrics which does well in both fluency and adequacy evaluation. It achieves the best results in fluency evaluation and comparable results to METEOR in adequacy evaluation, and the balanced performance leads to the best overall evaluation results in the experiment. To estimate the significance of the correlations, bootstrap resampling (Koehn, 2004) is used to randomly select 5514 sentences with replacement out of the whole test set of 5514 sentences, and then the correlation coefficients are computed based on the selected sentence set. The resampling is repeated 5000 times, and the 95% confidence intervals are shown in Tables 3, 4, and 5. We can see that it is very diffi-

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEU-5	BLEU-6	ROUGE-W
F	0.147	0.162	0.166	0.168	0.165	0.164	0.191
A	0.288	0.296	0.291	0.285	0.279	0.274	0.268
O	0.243	0.256	0.255	0.251	0.247	0.244	0.254

Table 1: Sentence level evaluation results of BLEU and ROUGE-W

	low	mean	high
B-3	(-16.6%) 0.138	0.165	0.192 (+16.4%)
R_1	(-17.8%) 0.124	0.151	0.177 (+17.3%)
R_2	(-14.3%) 0.164	0.191	0.218 (+14.2%)
M	(-15.8%) 0.139	0.166	0.191 (+15.5%)
S	(-13.3%) 0.174	0.201	0.227 (+13.3%)

Table 3: 95% significance intervals for sentence-level fluency evaluation

	low	mean	high
B-3	(-08.2%) 0.280	0.306	0.330 (+08.1%)
R_1	(-08.5%) 0.278	0.304	0.329 (+08.4%)
R_2	(-09.2%) 0.259	0.285	0.312 (+09.5%)
M	(-07.3%) 0.307	0.332	0.355 (+07.0%)
S	(-07.9%) 0.295	0.321	0.346 (+07.8%)

Table 4: 95% significance intervals for sentence-level adequacy evaluation

cult for one metric to significantly outperform another metric in sentence-level evaluation. The results show that the mean of the correlation factors converges right to the value we computed based on the whole testing set, and the confidence intervals correlate with the means.

While sentence-level evaluation is useful if we are interested in a confidence measure on MT outputs, syste-x level evaluation is more useful for comparing MT systems and guiding their development. Thus we also present the evaluation results based on the 7 MT output sets in Table 6. SIA uses the same decay factor as in the sentence-level evaluation. Its system-level score is computed as the arithmetic mean of the sentence level scores, and

	low	mean	high
B-3	(-09.8%) 0.238	0.264	0.290 (+09.9%)
R_1	(-10.2%) 0.229	0.255	0.281 (+10.0%)
R_2	(-10.0%) 0.238	0.265	0.293 (+10.4%)
M	(-09.0%) 0.254	0.279	0.304 (+08.8%)
S	(-08.7%) 0.265	0.291	0.316 (+08.8%)

Table 5: 95% significance intervals for sentence-level overall evaluation

	WLS	WLS PROB	WLS INCS	WLS PROB INCS
F	0.189	0.202	0.188	0.202
A	0.295	0.310	0.311	0.322
O	0.270	0.285	0.278	0.292

Table 7: Results of different components in SIA

	WLS INCS	WLS INCS STEM	WLS INCS WN	WLS INCS STEM WN
F	0.188	0.188	0.187	0.191
A	0.311	0.313	0.310	0.317
O	0.278	0.280	0.277	0.284

Table 8: Results of SIA working with Porter-Stem and WordNet

so are ROUGE, METEOR and the human judgments. We can see that SIA achieves the best performance in both fluency and adequacy evaluation of the 7 systems. Though the 7-sample based results are not reliable, we can get a sense of how well SIA works in the system-level evaluation.

4.3 Components in SIA

To see how the three components in SIA contribute to the final performance, we conduct experiments where one or two components are removed in SIA, shown in Table 7. The three components are denoted as WLS (weighted loose sequence alignment), PROB (stochastic word matching), and INCS (iterative alignment scheme) respectively. WLS without INCS does only one round of alignment and chooses the best alignment score as the final score. This scheme is similar to ROUGE-W and METEOR. We can see that INCS, as expected, improves the adequacy evaluation without hurting the fluency evaluation. PROB improves both adequacy and fluency evaluation performance. The result that SIA works with PORTER-STEM and WordNet is also shown in Table 8. When PORTER-STEM and WordNet are

	B-6	R_1	R_2	M	S
F	0.514	0.466	0.458	0.378	0.532
A	0.876	0.900	0.906	0.875	0.928
O	0.794	0.790	0.792	0.741	0.835

Table 6: Results of BLEU, ROUGE, METEOR and SIA in system level evaluation

both used, PORTER-STEM is used first. We can see that they are not as good as using the stochastic word matching. Since INCS and PROB are independent of WLS, we believe they can also be used to improve other metrics such as ROUGE-W and METEOR.

5 Conclusion

This paper describes a new metric SIA for MT evaluation, which achieves good performance by combining the advantages of n-gram-based metrics and loose-sequence-based metrics. SIA uses stochastic word mapping to allow soft or partial matches between the MT hypotheses and the references. This stochastic component is shown to be better than PORTER-STEM and WordNet in our experiments. We also analyzed the effect of other components in SIA and speculate that they can also be used in other metrics to improve their performance.

Acknowledgments This work was supported by NSF ITR IIS-09325646 and NSF ITR IIS-0428020.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judegments. In *Proceedings of the ACL-04 workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, Michigan.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Conference of the Association for Computational Linguistics (ACL-05)*.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-03)*, pages 16–23.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2003. Confidence estimation for machine translation. Technical report, Center for Language and Speech Processing, Johns Hopkins University, Baltimore. Summer Workshop Final Report.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

G. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *In HLT 2002, Human Language Technology Conference*, San Diego, CA.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388–395, Barcelona, Spain, July.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42th Annual Conference of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain.

Ding Liu and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. In *ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of the 2003 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-03)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02)*, Philadelphia, PA.