Source-Language Features and Maximum Correlation Training for Machine Translation Evaluation

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

Abstract

We propose three new features for MT evaluation: source-sentence constrained n-gram precision, source-sentence reordering metrics, and discriminative unigram precision, as well as a method of learning linear feature weights to directly maximize correlation with human judgments. By aligning both the hypothesis and the reference with the source-language sentence, we achieve better correlation with human judgments than previously proposed metrics. We further improve performance by combining individual evaluation metrics using maximum correlation training, which is shown to be better than the classification-based framework.

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. The most commonly used metric, BLEU, correlates well over large test sets with human judgments (Papineni et al., 2002), but does not perform as well on sentence-level evaluation (Blatz et al., 2003). Later approaches to improve sentence-level evaluation performance can be summarized as falling into four types:

- Metrics based on common loose sequences of MT outputs and references (Lin and Och, 2004; Liu and Gildea, 2006). Such metrics were shown to have better fluency evaluation performance than metrics based on n-grams such BLEU and NIST (Doddington, 2002).
- Metrics based on syntactic similarities such as the head-word chain metric (HWCM) (Liu and Gildea, 2005). Such metrics try to improve fluency evaluation performance for MT, but they heavily depend on automatic parsers, which are designed for well-formed sentences and cannot generate robust parse trees for MT outputs.
- Metrics based on word alignment between MT outputs and the references (Banerjee and Lavie, 2005). Such metrics do well in adequacy evaluation, but are not as good in fluency evaluation, because of their unigram basis (Liu and Gildea, 2006).
- Combination of metrics based on machine learning. Kulesza and Shieber (2004) used SVMs to combine several metrics. Their method is based on the assumption that higher classification accuracy in discriminating human- from machine-generated translations will yield closer correlation with human judgment. This assumption may not always hold, particularly when classification is difficult. Lita et al. (2005) proposed a log-linear model to combine features, but they only did preliminary experiments based on 2 features.

Following the track of previous work, to improve evaluation performance, one could either propose new metrics, or find more effective ways to combine the metrics. We explore both approaches. Much work has been done on computing MT scores based

Proceedings of NAACL HLT 2007, pages 41–48,
Rochester, NY, April 2007.

on the pair of MT output/reference, and we aim to investigate whether some other information could be used in the MT evaluation, such as source sentences. We propose two types of source-sentence related features as well as a feature based on part of speech. The three new types of feature can be summarized as follows:

- Source-sentence constrained n-gram precision. Overlapping n-grams between an MT hypothesis and its references do not necessarily indicate correct translation segments, since they could correspond to different parts of the source sentence. Thus our constrained n-gram precision counts only overlapping n-grams in MT hypothesis and reference which are aligned to the same words in the source sentences.

- Source-sentence reordering agreement. With the alignment information, we can compare the reorderings of the source sentence in the MT hypothesis and in its references. Such comparison only considers the aligned positions of the source words in MT hypothesis and references, and thus is oriented towards evaluating the sentence structure.

- Discriminative unigram precision. We divide the normal n-gram precision into many sub-precisions according to their part of speech (POS). The division gives us flexibility to train the weights of each sub-precision in frameworks such as SVM and Maximum Correlation Training, which will be introduced later. The motivation behind such differentiation is that different sub-precisions should have different importance in MT evaluation, e.g., sub-precision of nouns, verbs, and adjectives should be important for evaluating adequacy, and sub-precision in determiners and conjunctions should mean more in evaluating fluency.

Along the direction of feature combination, since indirect weight training using SVMs, based on reducing classification error, cannot always yield good performance, we train the weights by directly optimizing the evaluation performance, i.e., maximizing the correlation with the human judgment. This type of direct optimization is known as Minimum Error Rate Training (Och, 2003) in the MT community, and is an essential component in building the state-of-art MT systems. It would seem logical to apply similar methods to MT evaluation. What is more, Maximum Correlation Training (MCT) enables us to train the weights based on human fluency judgments and adequacy judgments respectively, and thus makes it possible to make a fluency-oriented or adequacy-oriented metric. It surpasses previous MT metrics' approach, where a a single metric evaluates both fluency and adequacy. The rest of the paper is organized as follows: Section 2 gives a brief recap of n-gram precision-based metrics and introduces our three extensions to them; Section 3 introduces MCT for MT evaluation; Section 4 describes the experimental results, and Section 5 gives our conclusion.

2 Three New Features for MT Evaluation

Since our source-sentence constrained n-gram precision and discriminative unigram precision are both derived from the normal n-gram precision, it is worth describing the original n-gram precision metric, BLEU (Papineni et al., 2002). For every MT hypothesis, BLEU computes the fraction of n-grams which also appear in the reference sentences, as well as a brevity penalty. The formula for computing BLEU is shown below:

$$\text{BLEU} = \frac{\text{BP}}{N} \sum_{n=1}^{N} \frac{\sum_C \sum_{ngram \in C} Count_{clip}(ngram)}{\sum_C \sum_{ngram' \in C} Count(ngram')}$$

where C denotes the set of MT hypotheses. $Count_{clip}(ngram)$ denotes the clipped number of n-grams in the candidates which also appear in the references. BP in the above formula denotes the brevity penalty, which is set to 1 if the accumulated length of the MT outputs is longer than the arithmetic mean of the accumulated length of the references, and otherwise is set to the ratio of the two. For sentence-level evaluation with BLEU, we compute the score based on each pair of MT hypothesis/reference. Later approaches, as described in Section 1, use different ways to manipulate the morphological similarity between the MT hypothesis and its references. Most of them, except NIST, consider the words in MT hypothesis as the same, i.e., as long as the words in MT hypothesis appear in the references,

they make no difference to the metrics.[1] NIST computes the n-grams weights as the logarithm of the ratio of the n-gram frequency and its one word lower n-gram frequency. From our experiments, NIST is not generally better than BLEU, and the reason, we conjecture, is that it differentiates the n-grams too much and the frequency estimated upon the evaluation corpus is not always reliable. In this section we will describe two other strategies for differentiating the n-grams, one of which uses the alignments with the source sentence as a further constraint, while the other differentiates the n-gram precisions according to POS.

2.1 Source-sentence Constrained N-gram Precision

The quality of an MT sentence should be independent of the source sentence given the reference translation, but considering that current metrics are all based on shallow morphological similarity of the MT outputs and the reference, without really understanding the meaning in both sides, the source sentences could have some useful information in differentiating the MT outputs. Consider the Chinese-English translation example below:

Source: *wo bu neng zhe me zuo*
Hypothesis: *I must hardly not do this*
Reference: *I must not do this*

It is clear that the word *not* in the MT output cannot co-exist with the word *hardly* while maintaining the meaning of the source sentence. None of the metrics mentioned above can prevent *not* from being counted in the evaluation, due to the simple reason that they only compute shallow morphological similarity. Then how could the source sentence help in the example? If we reveal the alignment of the source sentence with both the reference and the MT output, the Chinese word *bu neng* would be aligned to *must not* in the reference and *must hardly* in the MT output respectively, leaving the word *not* in the MT output not aligned to any word in the source sentence. Therefore, if we can somehow find the alignments between the source sentence and the reference/MT output, we could be smarter in selecting the overlapping words to be counted in the metric: only select the words which are aligned to the same source words. Now the question comes to how to find the alignment of source sentence and MT hypothesis/references, since the evaluation data set usually does not contain alignment information. Our approach uses GIZA++[2] to construct the many-to-one alignments between source sentences and the MT hypothesis/references respectively.[3] GIZA++ could generate many-to-one alignments either from source sentence to the MT hypothesis, in which case every word in MT hypothesis is aligned to a set of (or none) words in the source sentence, or from the reverse direction, in which case every word in MT hypothesis is aligned to exactly one word (or none) word in the source sentence. In either case, using $MTalign_i$ and $REFalign_i$ to denote the positions of the words in the source sentences which are aligned to a word in the MT hypothesis and a word in the reference respectively, the algorithm for computing source-sentence constrained n-gram precision of length n is described in Figure 1.

for all n-grams $w_i, ..., w_{i+n-1}$ in MT hypothesis **do**
  $max\_val = 0$;
  for all reference sentences **do**
    for all n-grams $r_j, ..., r_{j+n-1}$ in current reference sentence **do**
      val=0;
      for k=0; k $\leq$ n-1; k ++ **do**
        if w_{i+k} equals r_{j+k} AND $MTalign_i$ equals $REFalign_j$ **then**
          val += $\frac{1}{n}$;
      if $val \geq max\_val$ **then**
        $max\_val = val$;
  hit_count += max_val;
return $\frac{hit\_count}{MThypothesislength} \times length\_penalty$;

Figure 1: Algorithm for Computing Source-sentence Constrained n-gram Precision

Since source-sentence constrained n-gram precision (SSCN) is a precision-based metric, the vari-

[1] In metrics such as METEOR, ROUGE, SIA (Liu and Gildea, 2006), the positions of words do make difference, but it has nothing to do with the word itself.

[2] GIZA++ is available at http://www.fjoch.com/GIZA++.html

[3] More refined alignments could be got for source-hypothesis from the MT system, and for source-references by using manual proof-reading after the automatic alignment. Doing so, however, requires the MT system's cooperation and some costly human labor.

able $length\_penalty$ is used to avoid assigning a short MT hypothesis a high score, and is computed in the same way as BLEU. Note that in the algorithm for computing the precision of n-grams longer than one word, not all words in the n-grams should satisfy the source-sentence constraint. The reason is that the high order n-grams are already very sparse in the sentence-level evaluation. To differentiate the SSCNs based on the source-to-MT/Ref (many-to-one) alignments and the MT/Ref-to-source (many-to-one) alignments, we use SSCN1 and SSCN2 to denote them respectively. Naturally, we could combine the constraint in SSCN1 and SSCN2 by either taking their union (the combined constrained is satisfied if either one is satisfied) or intersecting them (the combined constrained is satisfied if both constraints are satisfied). We use SSCN_u and SSCN_i to denote the SSCN based on unioned constraints and intersected constraints respectively. We could also apply the stochastic word mapping proposed in SIA (Liu and Gildea, 2006) to replace the hard word matching in Figure 1, and the corresponding metrics are denoted as pSSCN1, pSSCN2, pSSCN_u, pSSCN_i, with the suffixed number denoting different constraints.

2.2 Metrics Based on Source Word Reordering

Most previous MT metrics concentrate on the co-occurrence of the MT hypothesis words in the references. Our metrics based on source sentence reorderings, on the contrary, do not take words identities into account, but rather compute how similarly the source words are reordered in the MT output and the references. For simplicity, we only consider the pairwise reordering similarity. That is, for the source word pair w_i and w_j, if their aligned positions in the MT hypothesis and a reference are in the same order, we call it a consistent word pair. Our pairwise reordering similarity (PRS) metric computes the fraction of the consistent word pairs in the source sentence. Figure 2 gives the formal description of PRS. $SrcMT_i$ and $SrcRef_{k,i}$ denote the aligned position of source word w_i in the MT hypothesis and the kth reference respectively, and N denotes the length of the source sentence.

Another criterion for evaluating the reordering of the source sentence in the MT hypothesis is how well it maintains the original word order in the source sentence. We know that most of the time, the alignment of the source sentence and the MT hypothesis is monotonic. This idea leads to the metric of monotonic pairwise ratio (MPR), which computes the fraction of the source word pairs whose aligned positions in the MT hypothesis are of the same order. It is described in Figure 3.

for all word pair w_i, w_j in the source sentence such that $i < j$ **do**
  for all reference sentences r_k **do**
    if ($SrcMT_i == SrcMT_j$ AND $SrcRef_{k,i} == SrcRef_{k,j}$) OR $((SrcMT_i - SrcMT_j) \times (SrcRef_{k,i} - SrcRef_{k,j}) > 0)$ **then**
      $count++$; break;
return $\frac{2 \times count}{N \times (N-1)}$;

Figure 2: Compute Pairwise Reordering Similarity

for all word pair w_i, w_j in the source sentence, such that $i < j$ **do**
  if $SrcMT_i - SrcMT_j < 0$ **then**
    $count++$;
return $\frac{2 \times count}{N \times (N-1)}$;

Figure 3: Compute Source Sentence Monotonic Reordering Ratio

2.3 Discriminative Unigram Precision Based on POS

The Discriminative Unigram Precision Based on POS (DUPP) decomposes the normal unigram precision into many sub-precisions according to their POS. The algorithm is described in Figure 4.

These sub-precisions by themselves carry the same information as standard unigram precision, but they provide us the opportunity to make a better combined metric than the normal unigram precision with MCT, which will be introduced in next section.

for all unigram *s* in the MT hypothesis **do**
  if *s* is found in any of the references **then**
    $count_{POS(s)}$ += 1

$$precision_x = \frac{count_x}{mt\ \ hypothesis\ \ length} \quad \forall x \in POS$$

Figure 4: Compute DUPP for N-gram with length n

Such division could in theory be generalized to work with higher order n-grams, but doing so would make the n-grams in each POS set much more sparse. The preprocessing step for the metric is tagging both the MT hypothesis and the references with POS. It might elicit some worries about the robustness of the POS tagger on the noise-containing MT hypothesis. This should not be a problem for two reasons. First, compared with other preprocessing steps like parsing, POS tagging is easier and has higher accuracy. Second, because the counts for each POS are accumulated, the correctness of a single word's POS will not affect the result very much.

3 Maximum Correlation Training for Machine Translation Evaluation

Maximum Correlation Training (MCT) is an instance of the general approach of directly optimizing the objective function by which a model will ultimately be evaluated. In our case, the model is the linear combination of the component metrics, the parameters are the weights for each component metric, and the objective function is the Pearson's correlation of the combined metric and the human judgments. The reason to use the linear combination of the metrics is that the component metrics are usually of the same or similar order of magnitude, and it makes the optimization problem easy to solve. Using w to denote the weights, and m to denote the component metrics, the combined metric x is computed as:

$$x(w) = \sum_j w_j m_j \tag{1}$$

Using h_i and $x(w)_i$ denote the human judgment and combined metric for a sentence respectively, and N denote the number of sentences in the evaluation set, the objective function is then computed as:

$$Pearson(X(w), H) = \frac{\sum_{i=1}^{N} x(w)_i h_i - \frac{\sum_{i=1}^{N} x(w)_i \sum_{i=1}^{N} h_i}{N}}{\sqrt{(\sum_{i=1}^{N} x(w)_i^2 - \frac{(\sum_{i=1}^{N} x(w)_i)^2}{N})(\sum_{i=1}^{N} h_i^2 - \frac{(\sum_{i=1}^{N} h_i)^2}{N})}}$$

Now our task is to find the weights for each component metric so that the correlation of the combined metric with the human judgment is maximized. It can be formulated as:

$$w = \arg\max_w Pearson(X(w), H) \tag{2}$$

The function $Pearson(X(w), H)$ is differentiable with respect to the vector w, and we compute this derivative analytically and perform gradient ascent. Our objective function not always convex (one can easily create a non-convex function by setting the human judgments and individual metrics to some particular value). Thus there is no guarantee that, starting from a random w, we will get the globally optimal w using optimization techniques such as gradient ascent. The easiest way to avoid ending up with a bad local optimum to run gradient ascent by starting from different random points. In our experiments, the difference in each run is very small, i.e., by starting from different random initial values of w, we end up with, not the same, but very similar values for Pearson's correlation.

4 Experiments

Experiments were conducted to evaluate the performance of the new metrics proposed in this paper, as well as the MCT combination framework. The 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), each of which contains 919 English sentences translated from the same set of Chinese sentences. There are four references (E01, E02, E03, E04) and two sets of human scores for each MT hypothesis. Each human score set contains a fluency and an adequacy score, both of which range from 1 to 5. We create a set of overall human scores by averaging the human fluency and adequacy scores. For evaluating the automatic metrics, we compute the Pearson's correlation of the automatic scores and the averaged human scores (over the two sets of available human scores), for overall score, fluency, and adequacy. The alignment between the source sentences and the MT hypothesis/references is computed by GIZA++, which is trained on the combined corpus of the evaluation data and a parallel corpus of Chinese-English newswire text. The parallel newswire corpus contains around 75,000 sentence pairs, 2,600,000 English words and 2,200,000 Chinese words. The

stochastic word mapping is trained on a French-English parallel corpus containing 700,000 sentence pairs, and, following Liu and Gildea (2005), we only keep the top 100 most similar words for each English word.

4.1 Performance of the Individual Metrics

To evaluate our source-sentence based metrics, they are used to evaluate the 7 MT outputs, with the 4 sets of human references. The sentence-level Pearson's correlation with human judgment is computed for each MT output, and the averaged results are shown in Table 1. As a comparison, we also show the results of BLEU, NIST, METEOR, ROUGE, WER, and HWCM. For METEOR and ROUGE, WORDNET and PORTER-STEMMER are enabled, and for SIA, the decay factor is set to 0.6. The number in brackets, for BLEU, shows the n-gram length it counts up to, and for SSCN, shows the length of the n-gram it uses. In the table, the top 3 results in each column are marked bold and the best result is also underlined. The results show that the SSCN2 metrics are better than the SSCN1 metrics in adequacy and overall score. This is understandable since what SSCN metrics need is which words in the source sentence are aligned to an n-gram in the MT hypothesis/references. This is directly modeled in the alignment used in SSCN2. Though we could also get such information from the reverse alignment, as in SSCN1, it is rather an indirect way and could contain more noise. It is interesting that SSCN1 gets better fluency evaluation results than SSCN2. The SSCN metrics with the unioned constraint, SSCN_u, by combining the strength of SSCN1 and SSCN2, get even better results in all three aspects. We can see that SSCN metrics, even without stochastic word mapping, get significantly better results than their relatives, BLEU, which indicates the source sentence constraints do make a difference. SSCN2 and SSCN_u are also competitive to the state-of-art MT metrics such as METEOR and SIA. The best SSCN metric, pSSCN_u(2), achieves the best performance among all the testing metrics in overall and adequacy, and the second best performance in fluency, which is just a little bit worse than the best fluency metric SIA.

The two reordering based metrics, PRS and MPR, are not as good as the other testing metrics, in terms of the individual performance. It should not be surprising since they are totally different kind of metrics, which do not count the overlapping n-grams, but the consistent/monotonic word pair reorderings. As long as they capture some property of the MT hypothesis, they might be able to boost the performance of the combined metric under the MCT framework.

	Fluency	Adequacy	Overall
ROUGE_W	24.8	27.8	29.0
ROUGE_S	19.7	30.9	28.5
METEOR	24.4	**34.8**	**33.1**
SIA	**26.8**	32.1	32.6
NIST_1	09.6	22.6	18.5
WER	22.5	27.5	27.7
PRS	14.2	19.4	18.7
MPR	11.0	18.2	16.5
BLEU(1)	18.4	29.6	27.0
BLEU(2)	20.4	31.1	28.9
BLEU(3)	20.7	30.4	28.6
HWCM(2)	22.1	30.3	29.2
SSCN1(1)	24.2	29.6	29.8
SSCN2(1)	22.9	33.0	31.3
SSCN_u(1)	23.8	34.2	32.5
SSCN_i(1)	23.4	28.0	28.5
pSSCN1(1)	24.9	30.2	30.6
pSSCN2(1)	23.8	34.0	32.4
pSSCN_u(1)	24.5	**34.6**	**33.1**
pSSCN_i(1)	24.1	28.8	29.3
SSCN1(2)	24.0	29.6	29.7
SSCN2(2)	23.3	31.5	31.8
SSCN_u(2)	24.1	34.5	**32.8**
SSCN_i(2)	23.1	27.8	28.2
pSSCN1(2)	**24.9**	30.2	30.6
pSSCN2(2)	24.3	34.4	**32.8**
pSSCN_u(2)	**25.2**	**35.4**	**33.9**
pSSCN_i(2)	23.9	28.7	29.1

Table 1: Performance of Component Metrics

4.2 Performance of the Combined Metrics

To test how well MCT works, the following scheme is used: each set of MT outputs is evaluated by MCT, which is trained on the other 6 sets of MT outputs and their corresponding human judgment; the averaged correlation of the 7 sets of MT outputs with the human judgment is taken as the final result.

4.2.1 Discriminative Unigram Precision based on POS

We first use MCT to combine the discriminative unigram precisions. To reduce the sparseness of the unigrams of each POS, we do not use the original POS set, but use a generalized one by combining

all POS tags with the same first letter (e.g., the different verb forms such as *VBN*, *VBD*, and *VBZ* are transformed to *V*). The unified POS set contains 23 POS tags. To give a fair comparison of DUPP with BLEU, the length penalty is also added into it as a component. Results are shown in Table 2. DUPP_f, DUPP_a and DUPP_o denote DUPP trained on human fluency, adequacy and overall judgment respectively. This shows that DUPP achieves obvious improvement over BLEU, with only the unigrams and length penalty, and DUPP_f/_a/_o gets the best result in fluency/adequacy/overall evaluation, showing that MCT is able to make a fluency- or adequacy-oriented metric.

4.2.2 Putting It All Together

The most interesting question in this paper is, with all these metrics, how well we can do in the MT evaluation. To answer the question, we put all the metrics described into the MCT framework and use the combined metric to evaluate the 7 MT outputs. Note that to speed up the training process, we do not directly use 24 DUPP components, instead, we use the 3 combined DUPP metrics. With the metrics shown in Table 1, we then have in total 31 metrics. Table 2 shows the results of the final combined metric. We can see that MCT trained on fluency, adequacy and overall human judgment get the best results among all the testing metrics in fluency, adequacy and overall evaluation respectively. We did a t-test with Fisher's z transform for the combined results and the individual results to see how significant the difference is. The combined results in adequacy and overall are significantly better at 99.5% confidence than the best results of the individual metrics (pSSCN_u(2)), and the combined result in fluency is significantly better at 96.9% confidence than the best individual metric (SIA). We also give the upper bound for each evaluation aspect by training MCT on the testing MT outputs, e.g., we train MCT on E09 and then use it to evaluate E09. The upper-bound is the best we can do with the MCT based on linear combination. Another linear framework, Classification SVM (CSVM),[4] is also used to combine the testing metrics except DUPP. Since DUPP is based on MCT, to make a neat comparison, we rule out DUPP in the experiments with CSVM. The testing scheme is the same as MCT, except that we only use 3 references for each MT hypothesis, and the positive samples for training CSVM are computed as the scores of one of the 4 references based on the other 3 references. The slack parameter of CSVM is chosen so as to maximize the classification accuracy of a heldout set of 800 negative and 800 positive samples, which are randomly selected from the training set. The results are shown in Table 2. We can see that MCT, with the same number of reference sentences, is better than CSVM. Note that the resources required by MCT and CSVM are different. MCT uses human judgments to adjust the weights, while CSVM needs extra human references to produce positive training samples.

	Fluency	Adequacy	Overall
DUPP_f	**23.6**	30.1	30.1
DUPP_a	22.1	**32.9**	30.9
DUPP_o	23.2	32.8	**31.3**
MCT_f(4)	**30.3**	36.7	37.2
MCT_a(4)	28.0	**38.9**	37.4
MCT_o(4)	29.4	38.8	**38.0**
Upper bound	35.3	43.4	42.2
MCT_f(3)	**29.2**	34.7	35.3
MCT_a(3)	27.4	**38.4**	36.8
MCT_o(3)	28.8	38.0	**37.2**
CSVM(3)	27.3	36.9	35.5

Table 2: Combination of the Testing Metrics

To have a rough idea of how the component metrics contribute to the final performance of MCT, we incrementally add metrics into the MCT in descending order of their overall evaluation performance, with the results shown in Figure 5. We can see that the performance improves as the number of metrics increases, in a rough sense. The major improvement happens in the 3rd, 4th, 9th, 14th, and 30th metrics, which are METEOR, SIA, DUPP_a, pSSCN1(1), and PRS. It is interesting to note that these are not the metrics with the highest individual performance. Another interesting observation is that there are no two metrics belonging to the same series in the most beneficial metrics, indicating that to get better combined metrics, individual metrics showing different sentence properties are preferred.

5 Conclusion

This paper first describes two types of new approaches to MT evaluation, which includes making

[4] http://svmlight.joachims.org/

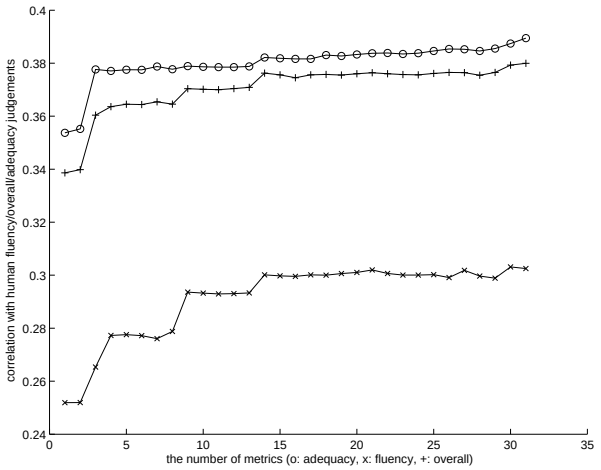

Figure 5: Performance as a Function of the Number of Interpolated Metrics

use of source sentences, and discriminating unigram precisions based on POS. Among all the testing metrics including BLEU, NIST, METEOR, ROUGE, and SIA, our new metric, pSSCN_u(2), based on source-sentence constrained bigrams, achieves the best adequacy and overall evaluation results, and the second best result in fluency evaluation. We further improve the performance by combining the individual metrics under the MCT framework, which is shown to be better than a classification based framework such as SVM. By examining the contribution of each component metric, we find that metrics showing different properties of a sentence are more likely to make a good combined metric.

Acknowledgments This work was supported by NSF grants IIS-0546554, IIS-0428020, and IIS-0325646.

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.

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.

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.

Alex Kulesza and Stuart M. Shieber. 2004. A learning approach to improving sentence-level MT evaluation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, Baltimore, MD, October.

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.

Lucian Vlad Lita, Monica Rogati, and Alon Lavie. 2005. Blanc: Learning evaluation metrics for mt. Vancouver.

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*.

Ding Liu and Daniel Gildea. 2006. Stochastic iterative alignment for machine translation evaluation. Sydney.

Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of ACL-03*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL-02*, Philadelphia, PA.