In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, July 21- 26, 2004.

Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics

Chin-Yew Lin and Franz Josef Och
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292, USA
{cyl,och}@isi.edu

Abstract

In this paper we describe two new objective automatic evaluation methods for machine translation. The first method is based on longest common subsequence between a candidate translation and a set of reference translations. Longest common subsequence takes into account sentence level structure similarity naturally and identifies longest co-occurring in-sequence n-grams automatically. The second method relaxes strict n-gram matching to skip-bigram matching. Skip-bigram is any pair of words in their sentence order. Skip-bigram co-occurrence statistics measure the overlap of skip-bigrams between a candidate translation and a set of reference translations. The empirical results show that both methods correlate with human judgments very well in both adequacy and fluency.

1 Introduction

Using objective functions to automatically evaluate machine translation quality is not new. Su et al. (1992) proposed a method based on measuring edit distance (Levenshtein 1966) between candidate and reference translations. Akiba et al. (2001) extended the idea to accommodate multiple references. Nießen et al. (2000) calculated the length-normalized edit distance, called word error rate (WER), between a candidate and multiple reference translations. Leusch et al. (2003) proposed a related measure called position-independent word error rate (PER) that did not consider word position, i.e. using bag-of-words instead. Instead of error measures, we can also use accuracy measures that compute similarity between candidate and reference translations in proportion to the number of common words between them as suggested by Melamed (1995). An n-gram co-occurrence measure, BLEU, proposed by Papineni et al. (2001) that calculates co-occurrence statistics based on n-gram overlaps have shown great potential. A variant of BLEU developed by NIST (2002) has been used in two recent large-scale machine translation evaluations.

Recently, Turian et al. (2003) indicated that standard accuracy measures such as recall, precision, and the F-measure can also be used in evaluation of machine translation. However, results based on their method, General Text Matcher (GTM), showed that unigram F-measure correlated best with human judgments while assigning more weight to higher n-gram ($n > 1$) matches achieved similar performance as Bleu. Since unigram matches do not distinguish words in consecutive positions from words in the wrong order, measures based on position-independent unigram matches are not sensitive to word order and sentence level structure. Therefore, systems optimized for these unigram-based measures might generate adequate but not fluent target language.

Since BLEU has been used to report the performance of many machine translation systems and it has been shown to correlate well with human judgments, we will explain BLEU in more detail and point out its limitations in the next section. We then introduce a new evaluation method called ROUGE-L that measures sentence-to-sentence similarity based on the longest common subsequence statistics between a candidate translation and a set of reference translations in Section 3. Section 4 describes another automatic evaluation method called ROUGE-S that computes skip-bigram co-occurrence statistics. Section 5 presents the evaluation results of ROUGE-L, and ROUGE-S and compare them with BLEU, GTM, NIST, PER, and WER in correlation with human judgments in terms of adequacy and fluency. We conclude this paper and discuss extensions of the current work in Section 6.

2 BLEU and N-gram Co-Occurrence

To automatically evaluate machine translations the machine translation community recently adopted an n-gram co-occurrence scoring procedure BLEU (Papineni et al. 2001). In two recent large-scale machine translation evaluations sponsored by NIST, a closely related automatic evaluation method, simply called NIST score, was used. The NIST (NIST 2002) scoring method is based on BLEU.

The main idea of BLEU is to measure the similarity between a candidate translation and a set of reference translations with a numerical metric. They used a weighted average of variable length n-gram matches between system translations and a set of human reference translations and showed that the weighted average metric correlating highly with human assessments.

BLEU measures how well a machine translation overlaps with multiple human translations using n-gram co-occurrence statistics. N-gram precision in BLEU is computed as follows:

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count(n\text{-}gram)} \quad (1)$$

Where $Count_{clip}$(*n-gram*) is the maximum number of *n-gram*s co-occurring in a candidate translation and a reference translation, and *Count*(*n-gram*) is the number of *n-gram*s in the candidate translation. To prevent very short translations that try to maximize their precision scores, BLEU adds a brevity penalty, *BP*, to the formula:

$$BP = \begin{Bmatrix} 1 & if\ |c| > |r| \\ e^{(1-|r|/|c|)} & if\ |c| \le |r| \end{Bmatrix} \quad (2)$$

Where |*c*| is the length of the candidate translation and |*r*| is the length of the reference translation. The BLEU formula is then written as follows:

$$BLEU = BP \bullet \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \quad (3)$$

The weighting factor, w_n, is set at 1/*N*.

Although BLEU has been shown to correlate well with human assessments, it has a few things that can be improved. First the subjective application of the brevity penalty can be replaced with a recall related parameter that is sensitive to reference length. Although brevity penalty will penalize candidate translations with low recall by a factor of $e^{(1-|r|/|c|)}$, it would be nice if we can use the traditional recall measure that has been a well known measure in NLP as suggested by Melamed (2003). Of course we have to make sure the resulting composite function of precision and recall is still correlates highly with human judgments.

Second, although BLEU uses high order n-gram (n>1) matches to favor candidate sentences with consecutive word matches and to estimate their fluency, it does not consider sentence level structure. For example, given the following sentences:

S1. police killed the gunman
S2. police kill the gunman[1]
S3. the gunman kill police

We only consider BLEU with unigram and bigram, i.e. *N*=2, for the purpose of explanation and call this BLEU-2. Using S1 as the reference and S2 and S3 as the candidate translations, S2 and S3 would have the same BLEU-2 score, since they both have one bigram and three unigram matches[2]. However, S2 and S3 have very different meanings.

Third, BLEU is a geometric mean of unigram to N-gram precisions. Any candidate translation without a N-gram match has a per-sentence BLEU score of zero. Although BLEU is usually calculated over the whole test corpus, it is still desirable to have a measure that works reliably at sentence level for diagnostic and introspection purpose.

To address these issues, we propose three new automatic evaluation measures based on longest common subsequence statistics and skip bigram co-occurrence statistics in the following sections.

3 Longest Common Subsequence

3.1 ROUGE-L

A sequence $Z = [z_1, z_2, ..., z_n]$ is a subsequence of another sequence $X = [x_1, x_2, ..., x_m]$, if there exists a strict increasing sequence $[i_1, i_2, ..., i_k]$ of indices of *X* such that for all $j = 1, 2, ..., k$, we have $x_{i_j} = z_j$ (Cormen et al. 1989). Given two sequences *X* and *Y*, the longest common subsequence (LCS) of *X* and *Y* is a common subsequence with maximum length. We can find the LCS of two sequences of length *m* and *n* using standard dynamic programming technique in *O*(*mn*) time.

LCS has been used to identify cognate candidates during construction of N-best translation lexicons from parallel text. Melamed (1995) used the ratio (LCSR) between the length of the LCS of two words and the length of the longer word of the two words to measure the cognateness between them. He used as an approximate string matching algorithm. Saggion et al. (2002) used normalized pairwise LCS (NP-LCS) to compare similarity between two texts in automatic summarization evaluation. NP-LCS can be shown as a special case of Equation (6) with $\beta = 1$. However, they did not provide the correlation analysis of NP-LCS with

[1] This is a real machine translation output.

[2] The “kill” in S2 or S3 does not match with “killed” in S1 in strict word-to-word comparison.

human judgments and its effectiveness as an automatic evaluation measure.

To apply LCS in machine translation evaluation, we view a translation as a sequence of words. The intuition is that the longer the LCS of two translations is, the more similar the two translations are. We propose using LCS-based F-measure to estimate the similarity between two translations X of length m and Y of length n, assuming X is a reference translation and Y is a candidate translation, as follows:

$$R_{lcs} = \frac{LCS(X,Y)}{m} \quad (4)$$

$$P_{lcs} = \frac{LCS(X,Y)}{n} \quad (5)$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \quad (6)$$

Where $LCS(X,Y)$ is the length of a longest common subsequence of X and Y, and $\beta = P_{lcs}/R_{lcs}$ when $\partial F_{lcs}/\partial R_{lcs} = \partial F_{lcs}/\partial P_{lcs}$. We call the LCS-based F-measure, i.e. Equation 6, ROUGE-L. Notice that ROUGE-L is 1 when $X = Y$ since $LCS(X,Y) = m$ or n; while ROUGE-L is zero when $LCS(X,Y) = 0$, i.e. there is nothing in common between X and Y. F-measure or its equivalents has been shown to have met several theoretical criteria in measuring accuracy involving more than one factor (Van Rijsbergen 1979). The composite factors are LCS-based recall and precision in this case. Melamed et al. (2003) used unigram F-measure to estimate machine translation quality and showed that unigram F-measure was as good as BLEU.

One advantage of using LCS is that it does not require consecutive matches but in-sequence matches that reflect sentence level word order as n-grams. The other advantage is that it automatically includes longest in-sequence common n-grams, therefore no predefined n-gram length is necessary. ROUGE-L as defined in Equation 6 has the property that its value is less than or equal to the minimum of unigram F-measure of X and Y. Unigram recall reflects the proportion of words in X (reference translation) that are also present in Y (candidate translation); while unigram precision is the proportion of words in Y that are also in X. Unigram recall and precision count all co-occurring words regardless their orders; while ROUGE-L counts only in-sequence co-occurrences.

By only awarding credit to in-sequence unigram matches, ROUGE-L also captures sentence level structure in a natural way. Consider again the example given in Section 2 that is copied here for convenience:

S1. police killed the gunman
S2. police kill the gunman
S3. the gunman kill police

As we have shown earlier, BLEU-2 cannot differentiate S2 from S3. However, S2 has a ROUGE-L score of 3/4 = 0.75 and S3 has a ROUGE-L score of 2/4 = 0.5, with $\beta = 1$. Therefore S2 is better than S3 according to ROUGE-L. This example also illustrated that ROUGE-L can work reliably at sentence level.

However, LCS only counts the main in-sequence words; therefore, other longest common subsequences and shorter sequences are not reflected in the final score. For example, consider the following candidate sentence:

S4. the gunman police killed

Using S1 as its reference, LCS counts either "the gunman" or "police killed", but not both; therefore, S4 has the same ROUGE-L score as S3. BLEU-2 would prefer S4 over S3. In Section 4, we will introduce skip-bigram co-occurrence statistics that do not have this problem while still keeping the advantage of in-sequence (not necessary consecutive) matching that reflects sentence level word order.

3.2 Multiple References

So far, we only demonstrated how to compute ROUGE-L using a single reference. When multiple references are used, we take the maximum LCS matches between a candidate translation, c, of n words and a set of u reference translations of m_j words. The LCS-based F-measure can be computed as follows:

$$R_{lcs\text{-}multi} = \max_{j=1}^{u}\left(\frac{LCS(r_j,c)}{m_j}\right) \quad (7)$$

$$P_{lcs\text{-}multi} = \max_{j=1}^{u}\left(\frac{LCS(r_j,c)}{n}\right) \quad (8)$$

$$F_{lcs\text{-}multi} = \frac{(1+\beta^2)R_{lcs-multi}P_{lcs-multi}}{R_{lcs-multi}+\beta^2 P_{lcs-multi}} \quad (9)$$

where $\beta = P_{lcs\text{-}multi}/R_{lcs\text{-}multi}$ when $\partial F_{lcs\text{-}multi}/\partial R_{lcs\text{-}multi} = \partial F_{lcs\text{-}multi}/\partial P_{lcs\text{-}multi}$.

This procedure is also applied to computation of ROUGE-S when multiple references are used. In the next section, we introduce the skip-bigram co-occurrence statistics. In the next section, we describe how to extend ROUGE-L to assign more credits to longest common subsequences with consecutive words.

3.3 ROUGE-W: Weighted Longest Common Subsequence

LCS has many nice properties as we have described in the previous sections. Unfortunately, the basic LCS also has a problem that it does not differentiate LCSes of different spatial relations within their embedding sequences. For example, given a reference sequence X and two candidate sequences Y_1 and Y_2 as follows:

X: [A B C D E F G]
Y_1: [A B C D H I K]
Y_2: [A H B K C I D]

Y_1 and Y_2 have the same ROUGE-L score. However, in this case, Y_1 should be the better choice than Y_2 because Y_1 has consecutive matches. To improve the basic LCS method, we can simply remember the length of consecutive matches encountered so far to a regular two dimensional dynamic program table computing LCS. We call this weighted LCS (WLCS) and use k to indicate the length of the current consecutive matches ending at words x_i and y_j. Given two sentences X and Y, the WLCS score of X and Y can be computed using the following dynamic programming procedure:

```
(1) For ($i = 0$; $i <= m$; $i$++)
        $c(i,j) = 0$ // initialize c-table
        $w(i,j) = 0$ // initialize w-table
(2) For ($i = 1$; $i <= m$; $i$++)
        For ($j = 1$; $j <= n$; $j$++)
          If $x_i = y_j$ Then
            // the length of consecutive matches at
            // position i-1 and j-1
            $k = w(i-1,j-1)$
            $c(i,j) = c(i-1,j-1) + f(k+1) - f(k)$
            // remember the length of consecutive
            // matches at position i, j
            $w(i,j) = k+1$
          Otherwise
            If $c(i-1,j) > c(i,j-1)$ Then
              $c(i,j) = c(i-1,j)$
              $w(i,j) = 0$        // no match at i, j
            Else $c(i,j) = c(i,j-1)$
              $w(i,j) = 0$        // no match at i, j
(3) $WLCS(X,Y) = c(m,n)$
```

Where c is the dynamic programming table, $c(i,j)$ stores the WLCS score ending at word x_i of X and y_j of Y, w is the table storing the length of consecutive matches ended at c table position i and j, and f is a function of consecutive matches at the table position, $c(i,j)$. Notice that by providing different weighting function f, we can parameterize the WLCS algorithm to assign different credit to consecutive in-sequence matches.

The weighting function f must have the property that $f(x+y) > f(x) + f(y)$ for any positive integers x and y. In other words, consecutive matches are awarded more scores than non-consecutive matches. For example, $f(k) = \alpha k - \beta$ when $k >= 0$, and $\alpha, \beta > 0$. This function charges a gap penalty of $-\beta$ for each non-consecutive n-gram sequences. Another possible function family is the polynomial family of the form k^{α} where $\alpha > 1$. However, in order to normalize the final ROUGE-W score, we also prefer to have a function that has a close form inverse function. For example, $f(k) = k^2$ has a close form inverse function $f^{-1}(k) = k^{1/2}$. F-measure based on WLCS can be computed as follows, given two sequences X of length m and Y of length n:

$$R_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(m)}\right) \qquad (10)$$

$$P_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(n)}\right) \qquad (11)$$

$$F_{wlcs} = \frac{(1+\beta^2)R_{wlcs}P_{wlcs}}{R_{wlcs} + \beta^2 P_{wlcs}} \qquad (12)$$

Where f^{-1} is the inverse function of f. We call the WLCS-based F-measure, i.e. Equation 12, ROUGE-W. Using Equation 12 and $f(k) = k^2$ as the weighting function, the ROUGE-W scores for sequences Y_1 and Y_2 are 0.571 and 0.286 respectively. Therefore, Y_1 would be ranked higher than Y_2 using WLCS. We use the polynomial function of the form k^{α} in the ROUGE evaluation package. In the next section, we introduce the skip-bigram co-occurrence statistics.

4 ROUGE-S: Skip-Bigram Co-Occurrence Statistics

Skip-bigram is any pair of words in their sentence order, allowing for arbitrary gaps. Skip-bigram co-occurrence statistics measure the overlap of skip-bigrams between a candidate translation and a set of reference translations. Using the example given in Section 3.1:

S1. police killed the gunman
S2. police kill the gunman
S3. the gunman kill police
S4. the gunman police killed

Each sentence has $C(4,2)$[3] = 6 skip-bigrams. For example, S1 has the following skip-bigrams:

[3] Combination: $C(4,2) = 4!/(2!*2!) = 6$.

("police killed", "police the", "police gunman", "killed the", "killed gunman", "the gunman")

S2 has three skip-bigram matches with S1 ("*police the*", "*police gunman*", "*the gunman*"), S3 has one skip-bigram match with S1 ("*the gunman*"), and S4 has two skip-bigram matches with S1 ("*police killed*", "*the gunman*"). Given translations *X* of length *m* and *Y* of length *n*, assuming *X* is a reference translation and *Y* is a candidate translation, we compute skip-bigram-based F-measure as follows:

$$R_{skip2} = \frac{SKIP2(X,Y)}{C(m,2)} \qquad (13)$$

$$P_{skip2} = \frac{SKIP2(X,Y)}{C(n,2)} \qquad (14)$$

$$F_{skip2} = \frac{(1+\beta^2)R_{skip2}P_{skip2}}{R_{skip2}+\beta^2 P_{skip2}} \qquad (15)$$

Where *SKIP2*(*X*,*Y*) is the number of skip-bigram matches between *X* and *Y*, $\beta = P_{skip2}/R_{skip2}$ when $\partial F_{skip2}/\partial R_{skip2} = \partial F_{skip2}/\partial P_{skip2}$, and *C* is the combination function. We call the skip-bigram-based F-measure, i.e. Equation 15, ROUGE-S.

Using Equation 15 with $\beta = 1$ and S1 as the reference, S2's ROUGE-S score is 0.5, S3 is 0.167, and S4 is 0.333. Therefore, S2 is better than S3 and S4, and S4 is better than S3. This result is more intuitive than using BLEU-2 and ROUGE-L. One advantage of skip-bigram vs. BLEU is that it does not require consecutive matches but is still sensitive to word order. Comparing skip-bigram with LCS, skip-bigram counts all in-order matching word pairs while LCS only counts one longest common subsequence.

We can limit the maximum skip distance, d_{skip}, between two in-order words that is allowed to form a skip-bigram. Applying such constraint, we limit skip-bigram formation to a fix window size. Therefore, computation time can be reduced and hopefully performance can be as good as the version without such constraint. For example, if we set d_{skip} to 0 then ROUGE-S is equivalent to bigram overlap. If we set d_{skip} to 4 then only word pairs of at most 4 words apart can form skip-bigrams.

Adjusting Equations 13, 14, and 15 to use maximum skip distance limit is straightforward: we only count the skip-bigram matches, *SKIP2*(*X*,*Y*), within the maximum skip distance and replace denominators of Equations 13, *C*(*m*,2), and 14, *C*(*n*,2), with the actual numbers of within distance skip-bigrams from the reference and the candidate respectively.

In the next section, we present the evaluations of ROUGE-L, ROUGE-S, and compare their performance with other automatic evaluation measures.

5 Evaluations

One of the goals of developing automatic evaluation measures is to replace labor-intensive human evaluations. Therefore the first criterion to assess the usefulness of an automatic evaluation measure is to show that it correlates highly with human judgments in different evaluation settings. However, high quality large-scale human judgments are hard to come by. Fortunately, we have access to eight MT systems' outputs, their human assessment data, and the reference translations from 2003 NIST Chinese MT evaluation (NIST 2002a). There were 919 sentence segments in the corpus. We first computed averages of the adequacy and fluency scores of each system assigned by human evaluators. For the input of automatic evaluation methods, we created three evaluation sets from the MT outputs:

1. Case set: The original system outputs with case information.
2. NoCase set: All words were converted into lower case, i.e. no case information was used. This set was used to examine whether human assessments were affected by case information since not all MT systems generate properly cased output.
3. Stem set: All words were converted into lower case and stemmed using the Porter stemmer (Porter 1980). Since ROUGE computed similarity on surface word level, stemmed version allowed ROUGE to perform more lenient matches.

To accommodate multiple references, we use a Jackknifing procedure. Given N references, we compute the best score over N sets of N-1 references. The final score is the average of the N best scores using N different sets of N-1 references. The Jackknifing procedure is adopted since we often need to compare system and human performance and the reference translations are usually the only human translations available. Using this procedure, we are able to estimate average human performance by averaging N best scores of one reference vs. the rest N-1 references.

We then computed average BLEU1-12[4], GTM with exponents of 1.0, 2.0, and 3.0, NIST, WER, and PER scores over these three sets. Finally we applied ROUGE-L, ROUGE-W with weighting function $k^{1.2}$, and ROUGE-S without skip distance

[4] BLEUN computes BLEU over n-grams up to length N. Only BLEU1, BLEU4, and BLEU12 are shown in Table 1.

Adequacy	With Case Information (Case)						Lower Case (NoCase)						Lower Case & Stemmed (Stem)					
Method	P	95%L	95%U	S	95%L	95%U	P	95%L	95%U	S	95%L	95%U	P	95%L	95%U	S	95%L	95%U
BLEU1	0.86	0.83	0.89	0.80	0.71	0.90	0.87	0.84	0.90	0.76	0.67	0.89	0.91	0.89	0.93	0.85	0.76	0.95
BLEU4	0.77	0.72	0.81	0.77	0.71	0.89	0.79	0.75	0.82	0.67	0.55	0.83	0.82	0.78	0.85	0.76	0.67	0.89
BLEU12	0.66	0.60	0.72	0.53	0.44	0.65	0.72	0.57	0.81	0.65	0.25	0.88	0.72	0.58	0.81	0.66	0.28	0.88
NIST	0.89	0.86	0.92	0.78	0.71	0.89	0.87	0.85	0.90	0.80	0.74	0.92	0.90	0.88	0.93	0.88	0.83	0.97
WER	0.47	0.41	0.53	0.56	0.45	0.74	0.43	0.37	0.49	0.66	0.60	0.82	0.48	0.42	0.54	0.66	0.60	0.81
PER	0.67	0.62	0.72	0.56	0.48	0.75	0.63	0.58	0.68	0.67	0.60	0.83	0.72	0.68	0.76	0.69	0.62	0.86
ROUGE-L	0.87	0.84	0.90	0.84	0.79	0.93	0.89	0.86	0.92	0.84	0.71	0.94	0.92	0.90	0.94	0.87	0.76	0.95
ROUGE-W	0.84	0.81	0.87	0.83	0.74	0.90	0.85	0.82	0.88	0.77	0.67	0.90	0.89	0.86	0.91	0.86	0.76	0.95
ROUGE-S*	0.85	0.81	0.88	0.83	0.76	0.90	0.90	0.88	0.93	0.82	0.70	0.92	0.95	0.93	0.97	0.85	0.76	0.94
ROUGE-S0	0.82	0.78	0.85	0.82	0.71	0.90	0.84	0.81	0.87	0.76	0.67	0.90	0.87	0.84	0.90	0.82	0.68	0.90
ROUGE-S4	0.82	0.78	0.85	0.84	0.79	0.93	0.87	0.85	0.90	0.83	0.71	0.90	0.92	0.90	0.94	0.84	0.74	0.93
ROUGE-S9	0.84	0.80	0.87	0.84	0.79	0.92	0.89	0.86	0.92	0.84	0.76	0.93	0.94	0.92	0.96	0.84	0.76	0.94
GTM10	0.82	0.79	0.85	0.79	0.74	0.83	0.91	0.89	0.94	0.84	0.79	0.93	0.94	0.92	0.96	0.84	0.79	0.92
GTM20	0.77	0.73	0.81	0.76	0.69	0.88	0.79	0.76	0.83	0.70	0.55	0.83	0.83	0.79	0.86	0.80	0.67	0.90
GTM30	0.74	0.70	0.78	0.73	0.60	0.86	0.74	0.70	0.78	0.63	0.52	0.79	0.77	0.73	0.81	0.64	0.52	0.80

Fluency	With Case Information (Case)						Lower Case (NoCase)						Lower Case & Stemmed (Stem)					
Method	P	95%L	95%U	S	95%L	95%U	P	95%L	95%U	S	95%L	95%U	P	95%L	95%U	S	95%L	95%U
BLEU1	0.81	0.75	0.86	0.76	0.62	0.90	0.73	0.67	0.79	0.70	0.62	0.81	0.70	0.63	0.77	0.79	0.67	0.90
BLEU4	0.86	0.81	0.90	0.74	0.62	0.86	0.83	0.78	0.88	0.68	0.60	0.81	0.83	0.78	0.88	0.70	0.62	0.81
BLEU12	0.87	0.76	0.93	0.66	0.33	0.79	0.93	0.81	0.97	0.78	0.44	0.94	0.93	0.84	0.97	0.80	0.49	0.94
NIST	0.81	0.75	0.87	0.74	0.62	0.86	0.70	0.64	0.77	0.68	0.60	0.79	0.68	0.61	0.75	0.77	0.67	0.88
WER	0.69	0.62	0.75	0.68	0.57	0.85	0.59	0.51	0.66	0.70	0.57	0.82	0.60	0.52	0.68	0.69	0.57	0.81
PER	0.79	0.74	0.85	0.67	0.57	0.82	0.68	0.60	0.73	0.69	0.60	0.81	0.70	0.63	0.76	0.65	0.57	0.79
ROUGE-L	0.83	0.77	0.88	0.80	0.67	0.90	0.76	0.69	0.82	0.79	0.64	0.90	0.73	0.66	0.80	0.78	0.67	0.90
ROUGE-W	0.85	0.80	0.90	0.79	0.63	0.90	0.78	0.73	0.84	0.72	0.62	0.83	0.77	0.71	0.83	0.78	0.67	0.90
ROUGE-S*	0.84	0.78	0.89	0.79	0.62	0.90	0.80	0.74	0.86	0.77	0.64	0.90	0.78	0.71	0.84	0.79	0.69	0.90
ROUGE-S0	0.87	0.81	0.91	0.78	0.62	0.90	0.83	0.78	0.88	0.71	0.62	0.82	0.82	0.77	0.88	0.76	0.62	0.90
ROUGE-S4	0.84	0.79	0.89	0.80	0.67	0.90	0.82	0.77	0.87	0.78	0.64	0.90	0.81	0.75	0.86	0.79	0.67	0.90
ROUGE-S9	0.84	0.79	0.89	0.80	0.67	0.90	0.81	0.76	0.87	0.79	0.69	0.90	0.79	0.73	0.85	0.79	0.69	0.90
GTM10	0.73	0.66	0.79	0.76	0.60	0.87	0.71	0.64	0.78	0.80	0.67	0.90	0.66	0.58	0.74	0.80	0.64	0.90
GTM20	0.86	0.81	0.90	0.80	0.67	0.90	0.83	0.77	0.88	0.69	0.62	0.81	0.83	0.77	0.87	0.74	0.62	0.89
GTM30	0.87	0.81	0.91	0.79	0.67	0.90	0.83	0.77	0.87	0.73	0.62	0.83	0.83	0.77	0.88	0.71	0.60	0.83

Table 1. Pearson's ρ and Spearman's ρ correlations of automatic evaluation measures vs. **adequacy** and **fluency**: BLEU1, 4, and 12 are BLEU with maximum of 1, 4, and 12 grams, NIST is the NIST score, ROUGE-L is LCS-based F-measure (β = 1), ROUGE-W is weighted LCS-based F-measure (β = 1). ROUGE-S* is skip-bigram-based co-occurrence statistics with any skip distance limit, ROUGE-SN is skip-bigram-based F-measure (β = 1) with maximum skip distance of N, PER is position independent word error rate, and WER is word error rate. GTM 10, 20, and 30 are general text matcher with exponents of 1.0, 2.0, and 3.0. (Note, only BLEU1, 4, and 12 are shown here to preserve space.)

limit and with skip distant limits of 0, 4, and 9. Correlation analysis based on two different correlation statistics, Pearson's ρ and Spearman's ρ, with respect to adequacy and fluency are shown in Table 1.

The Pearson's correlation coefficient[5] measures the strength and direction of a *linear* relationship between any two variables, i.e. automatic metric score and human assigned mean coverage score in our case. It ranges from +1 to -1. A correlation of 1 means that there is a perfect positive linear relationship between the two variables, a correlation of -1 means that there is a perfect negative linear relationship between them, and a correlation of 0 means that there is no linear relationship between them. Since we would like to use automatic evaluation metric not only in comparing systems but also in in-house system development, a good linear correlation with human judgment would enable us to use automatic scores to predict corresponding human judgment scores. Therefore, Pearson's correlation coefficient is a good measure to look at.

Spearman's correlation coefficient[6] is also a measure of correlation between two variables. It is a non-parametric measure and is a special case of the Pearson's correlation coefficient when the values of data are converted into ranks before computing the coefficient. Spearman's correlation coefficient does not assume the correlation between the variables is linear. Therefore it is a useful correlation indicator even when good linear correlation, for example, according to Pearson's correlation coefficient between two variables could

[5] For a quick overview of the Pearson's coefficient, see: http://davidmlane.com/hyperstat/A34739.html.

[6] For a quick overview of the Spearman's coefficient, see: http://davidmlane.com/hyperstat/A62436.html.

not be found. It also suits the NIST MT evaluation scenario where multiple systems are ranked according to some performance metrics.

To estimate the significance of these correlation statistics, we applied bootstrap resampling, generating random samples of the 919 different sentence segments. The lower and upper values of 95% confidence interval are also shown in the table. Dark (green) cells are the best correlation numbers in their categories and light gray cells are statistically equivalent to the best numbers in their categories. Analyzing all runs according to the adequacy and fluency table, we make the following observations:

Applying the stemmer achieves higher correlation with adequacy but keeping case information achieves higher correlation with fluency except for BLEU7-12 (only BLEU12 is shown). For example, the Pearson's ρ (P) correlation of ROUGE-S* with adequacy increases from 0.85 (Case) to 0.95 (Stem) while its Pearson's ρ correlation with fluency drops from 0.84 (Case) to 0.78 (Stem). We will focus our discussions on the Stem set in adequacy and Case set in fluency.

The Pearson's ρ correlation values in the Stem set of the Adequacy Table, indicates that ROUGE-L and ROUGE-S with a skip distance longer than 0 correlate highly and linearly with adequacy and outperform BLEU and NIST. ROUGE-S* achieves that best correlation with a Pearson's ρ of 0.95. Measures favoring consecutive matches, i.e. BLEU4 and 12, ROUGE-W, GTM20 and 30, ROUGE-S0 (bigram), and WER have lower Pearson's ρ. Among them WER (0.48) that tends to penalize small word movement is the worst performer. One interesting observation is that longer BLEU has lower correlation with adequacy.

Spearman's ρ values generally agree with Pearson's ρ but have more equivalents.

The Pearson's ρ correlation values in the Stem set of the Fluency Table, indicates that BLEU12 has the highest correlation (0.93) with fluency. However, it is statistically indistinguishable with 95% confidence from all other metrics shown in the Case set of the Fluency Table except for WER and GTM10.

GTM10 has good correlation with human judgments in adequacy but not fluency; while GTM20 and GTM30, i.e. GTM with exponent larger than 1.0, has good correlation with human judgment in fluency but not adequacy.

ROUGE-L and ROUGE-S*, 4, and 9 are good automatic evaluation metric candidates since they perform as well as BLEU in fluency correlation analysis and outperform BLEU4 and 12 significantly in adequacy. Among them, ROUGE-L is the best metric in both adequacy and fluency correlation with human judgment according to Spearman's correlation coefficient and is statistically indistinguishable from the best metrics in both adequacy and fluency correlation with human judgment according to Pearson's correlation coefficient.

6 Conclusion

In this paper we presented two new objective automatic evaluation methods for machine translation, ROUGE-L based on longest common subsequence (LCS) statistics between a candidate translation and a set of reference translations. Longest common subsequence takes into account sentence level structure similarity naturally and identifies longest co-occurring in-sequence n-grams automatically while this is a free parameter in BLEU.

To give proper credit to shorter common sequences that are ignored by LCS but still retain the flexibility of non-consecutive matches, we proposed counting skip bigram co-occurrence. The skip-bigram-based ROUGE-S* (without skip distance restriction) had the best Pearson's ρ correlation of 0.95 in adequacy when all words were lower case and stemmed. ROUGE-L, ROUGE-W, ROUGE-S*, ROUGE-S4, and ROUGE-S9 were equal performers to BLEU in measuring fluency. However, they have the advantage that we can apply them on sentence level while longer BLEU such as BLEU12 would not differentiate any sentences with length shorter than 12 words (i.e. no 12-gram matches). We plan to explore their correlation with human judgments on sentence-level in the future.

We also confirmed empirically that adequacy and fluency focused on different aspects of machine translations. Adequacy placed more emphasis on terms co-occurred in candidate and reference translations as shown in the higher correlations in Stem set than Case set in Table 1; while the reverse was true in the terms of fluency.

The evaluation results of ROUGE-L, ROUGE-W, and ROUGE-S in machine translation evaluation are very encouraging. However, these measures in their current forms are still only applying string-to-string matching. We have shown that better correlation with adequacy can be reached by applying stemmer. In the next step, we plan to extend them to accommodate synonyms and paraphrases. For example, we can use an existing thesaurus such as WordNet (Miller 1990) or creating a customized one by applying automated synonym set discovery methods (Pantel and Lin 2002) to identify potential synonyms. Paraphrases can also be automatically acquired using statistical methods as shown by Barzilay and Lee (2003). Once we have acquired synonym and paraphrase data, we then need to design a soft matching function that assigns partial credits to these approximate matches. In this scenario, statistically generated data has the advantage of being able to provide scores reflecting the strength of similarity between synonyms and paraphrased.

ROUGE-L, ROUGE-W, and ROUGE-S have also been applied in automatic evaluation of summarization and achieved very promising results (Lin 2004). In Lin and Och (2004), we proposed a framework that automatically evaluated automatic MT evaluation metrics using only manual translations without further human involvement. According to the results reported in that paper, ROUGE-L, ROUGE-W, and ROUGE-S also outperformed BLEU and NIST.

References

Akiba, Y., K. Imamura, and E. Sumita. 2001. Using Multiple Edit Distances to Automatically Rank Machine Translation Output. In *Proceedings of the MT Summit VIII*, Santiago de Compostela, Spain.

Barzilay, R. and L. Lee. 2003. Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignmen. In *Proceeding of NAACL-HLT 2003*, Edmonton, Canada.

Leusch, G., N. Ueffing, and H. Ney. 2003. A Novel String-to-String Distance Measure with Applications to Machine Translation Evaluation. In *Proceedings of MT Summit IX*, New Orleans, U.S.A.

Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*.

Lin, C.Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Text Summarization Branches Out*, post-conference workshop of ACL 2004, Barcelona, Spain.

Lin, C.-Y. and F. J. Och. 2004. ORANGE: a Method for Evaluating Automatic Evaluation Metrics for Machine Translation. In *Proceedings of 20th International Conference on Computational Linguistic* (COLING 2004), Geneva, Switzerland.

Miller, G. 1990. WordNet: An Online Lexical Database. *International Journal of Lexicography*, 3(4).

Melamed, I.D. 1995. Automatic Evaluation and Uniform Filter Cascades for Inducing N-best Translation Lexicons. In *Proceedings of the 3rd Workshop on Very Large Corpora (WVLC3)*. Boston, U.S.A.

Melamed, I.D., R. Green and J. P. Turian. 2003. Precision and Recall of Machine Translation. In *Proceedings of NAACL/HLT 2003*, Edmonton, Canada.

Nießen S., F.J. Och, G, Leusch, H. Ney. 2000. An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, Athens, Greece.

NIST. 2002. Automatic Evaluation of Machine Translation Quality using N-gram Co-Occurrence Statistics. http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf

Pantel, P. and Lin, D. 2002. Discovering Word Senses from Text. In *Proceedings of SIGKDD-02*. Edmonton, Canada.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2001. BLEU: a Method for Automatic Evaluation of Machine Translation. *IBM Research Report RC22176 (W0109-022).*

Porter, M.F. 1980. An Algorithm for Suffix Stripping. *Program*, 14, pp. 130-137.

Saggion H., D. Radev, S. Teufel, and W. Lam. 2002. Meta-Evaluation of Summaries in a Cross-Lingual Environment Using Content-Based Metrics. In *Proceedings of COLING-2002*, Taipei, Taiwan.

Su, K.-Y., M.-W. Wu, and J.-S. Chang. 1992. A New Quantitative Quality Measure for Machine Translation System. In *Proceedings of COLING-92*, Nantes, France.

Thompson, H. S. 1991. Automatic Evaluation of Translation Quality: Outline of Methodology and Report on Pilot Experiment. In *Proceedings of the Evaluator's Forum*, ISSCO, Geneva, Switzerland.

Turian, J. P., L. Shen, and I. D. Melamed. 2003. Evaluation of Machine Translation and its Evaluation. In *Proceedings of MT Summit IX*, New Orleans, U.S.A.

Van Rijsbergen, C.J. 1979. *Information Retrieval*. Butterworths. London.