# Structured Common Subsequences for Automatic Machine Translation Evaluation

Chenchen Ding, Masao Utiyama, Eiichiro Sumita
ASTREC, NICT, Japan

{chenchen.ding, mutiyama, eiichiro.sumita}@nict.go.jp

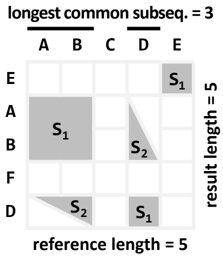
## 1 Introduction

To automatically evaluate the performance of a machine translation (MT) system is no less difficult a task than the automatic translation task itself. The most widely used measure in research community is the BLEU score [1], which can be interpreted as an arithmetic mean of the precision on $N$-gram matching with a brevity penalty, where $N$ usually is up to 4 in practice. As the $N$-grams used in BLEU are local features, which cannot reveal the structured information in transltion, metrics applying word order features are proposed. A typical one is RIBES [2], where word order is measured by Kendall's $\tau$ and combined with weighted uni-gram precisions. RIBES has shown especially high correlation with human evaluation in translation tasks requiring large amount of reordering operations, e.g., translation of long sentences between English and Japanese. However, BLEU is still a more human-correlated measure on those translation tasks not requiring reordering operation so heavily.

In our opinion, how to intuitively combine the local features and global structural features is an important issue in automatic evaluation metrics of MT. In BLEU, as mentioned, there is no structured features used, while in RIBES, the two kinds of features are combined with specific weights. Once such hyper-parameters are introduced, metrics then easily turn task-specific, with loss of generality at a certain level. Moreover, the meaning of hyper-parameters are usually hard to interpret. This is an crucial reason on why BLEU is so widely used despite its obvious defects. As there is no hyper-parameters,[1] BLEU is quite easy and intuitive to interpret, say, even if BLEU has only a mediocre correlation with human evaluation, at least we know a high BLEU score means a high precision on $N$-gram matching.

In this study, we proposed a measure based on twice common subsequence matching operation, referred to as *double common subsequence* (`dcs`) score. The first matching provides a component `cs1`, which

[1] Actually, the largest order of $N$-grams is a parameter, but a fixed 4 is overwhelmingly adopted.




$S$ = reference length $\times$ result length

$$cs_0 = \frac{\text{longest common subseq.}}{S} = \frac{3}{5}$$

$$cs_1 = \sqrt{\frac{\sum S_1}{S}} = \frac{\sqrt{6}}{5} \quad cs_2 = \sqrt{\frac{\sum S_2}{S}} = \frac{\sqrt{2}}{5}$$

$$dcs = \sqrt{\frac{\sum S_1 + \sum S_2}{S}} = \frac{\sqrt{8}}{5}$$

Figure 1: Taking `ABCDE` as a reference string, several metrics for the string of `EABFD`. `cs0` is a simple normalized length of longest common subsequence. `cs1`, `cs2` and `dcs` can be intuitively interpreted as the square root of the ratio of corresponding areas. ($S_2$ are depicted symmetrically as two triangles to indicate it is a kind of "monotone bonus" for subsequences `AB` and `D`)

reveals the local features by the length of matched subsequences, based on which a second-order matching provides a component `cs2`, showing the monotone tendency among the matched subsequences. The `dcs` score is a geometric mean of `cs1` and `cs2`. Details of the proposed `dcs` will be described in Sec. 3 and an example is shown in Fig. 1 for intuitive understanding. Basically, the `dcs` is still a measure based on matching of MT output and reference by human translation, like BLEU and RIBES. However, we designed it with special merits as follows.

1. no parameter needed to be tuned
2. applicable on character-level for languages without word separators, e.g., Chinese and Japanese
3. local and structured features combined naturally

We evaluate the proposed measure on translation tasks at *Workshop on Asian Translation* (WAT) [3, 4, 5, 6]. We have found that `cs2` component outperforms RIBES on heavily reordering-required tasks as English-to-Japanese and Japanese-to-English translations, as well as monotonous translation task of Korean-to-Japanese. On tasks requiring medium reordering such as Chinese-to-Japanese and Japanese-to-Chinese translation, the `dcs` score can provide a stable performance comparable with BLEU, without the affects introduced by tokenizing.

## 2 Related Work

Besides the mentioned BLEU and RIBES, various metrics for MT evaluation have been proposed. A simple and widely used category is to compare one or more references translated by human with the output of an automatic MT system. The BLEU and RIBES are both belong to this category. Other metrics in this category including editing distance based measures such as TER [7], and a pack of recall based measures called ROUGE [8] (or ORANGE [9], for several common sequence based metrics). More sophisticated metrics may rely on linguistic analysis where a typical measure is METEOR [10, 11]. Metrics in this category generally require well prepared monolingual data. Another line of research is reference-free evaluation, where evaluation models are trained from parallel data. A recently proposed measure is AMFM [12]. This kind of metrics can capture deeper semantic features but requires considerable parallel data to train numerous parameters.

## 3 Proposed Method

The proposed method is inspired by ROUGE-L [8], which is a measure by longest common subsequences (`LCS`) between reference and MT output. As addressed, the combination of local and structured features is an important issue for evaluation metrics. The `LCS` can thus combine the two kinds of features to a certain extent. However, there are still limitations by simple `LCS` as, 1) the granularity of matched strings are not addressed enough and 2) the non-monotone matched parts are not taken into consideration. For 1), an improved ROUGE-W [8] has been proposed to weight longer continuous subsequences more, while 2) is an intrinsic problem of `LCS` that it can only choose one path in matching, and features not lying on the optimal path are all neglected.

We improve `LCS`-based methods in two steps. First, rather than the optimal longest path in `LCS` is considered, we take all *non-nested common subsequences* into consideration. That is, we collect all local continuous matched subsequences no matter the relative order among them. Taking Fig. 1 as an example, there are three such subsequences of `AB`,[2] `D`, and `E`. Notice the subsequence of `E` will be omitted if only the `LCS` (i.e., `ABD`) is considered. Such a set of common subsequences can provide a more complete set of local matching features, while the structured information contained in `LCS` is lost. Therefore, a second common subsequence matching is conducted to filter out the monotonous parts within the results of the first matching. The basic unit in the second matching is the subsequences obtained in the first matching, e.g., the (`AB`, `D`) pair is identified as a monotonous pair in Fig. 1. Hence, the structured information lost in the first matching is brought in by the second matching.[3]

Based on the two steps of matching, we can calculate two components for local and structured features respectively. Inspired by the weighting method in ROUGE-W and intuitiveness, we set the overall framework in a quadratic manner. As shown in Fig. 1, the `cs1` component for local features is the square root of the summation over the squared lengths for all subsequences in the first matching; the `cs2` component for structured features is the square root of the summation over terms proportional to the lengths of neighboring monotonous subsequence identified in the second matching; and an overall `dcs` is equal to $\sqrt{\texttt{cs1}^2 + \texttt{cs2}^2}$. Considering two common subsequences with lengths of $x$ and $y$, because $(x+y)^2 > x^2 + xy + y^2 > x^2 + y^2$ is always true when $x > 0,\ y > 0$, `dcs` most appreciates long and continuous common subsequences (i.e., when the two subsequences are concatenated to one with a length of $(x+y)$), and moderately appreciates common subsequences with a monotonous order (i.e., there is an extra $xy$ term added to $x^2+y^2$, which, after all, cannot be larger than $(x+y)^2$).[4] The `Python` codes[5] of our implementation of `dcs` is presented in Table 1.

## 4 Evaluation

We selected tasks having no less than ten attending teams at WAT and calculated the Pearson's $\rho$ between the human-evaluation score and automatic metrics. The official evaluation metrics in WAT are BLEU, RIBES, and AMFM. The BLEU and RIBES for Chinese and Japanese are based on different tokenizers, while AMFM are character-based. For tasks with the two languages as target language, we also added character-based BLEU of 4- and 8-grams in comparison. As to the proposed methods using common subsequences, we first tested the simplest `LCS`-based measure, i.e., `cs0` in Fig. 1. This measure is essentially identical to ROUGH-L but the normalization way on sentence length is slightly different. The `dcs` and its components of `cs1` and `cs2` are evaluated respectively. All the common subsequence based measures were conducted on character-level for Chinese and Japanese. Text normalization on digits and punctuation marks were according to the instruction of the WAT.

[2] `A` and `B` are also common subsequences while they are the nested parts of `AB`.

[3] The second step matching may provide more structured features than `LCS`, because it is possible to have monotonous subsequences away from the optimal path.

[4] If the $xy$ term is weighted in $(0, 2)$, the relation is still true and we will have a `dcs` with weighted `cs1` and `cs2`.

[5] Executable under `Python 2.x`. As the `filter ()` does not return a list in `Python 3.x`, slight modification will be needed.

```
def nosub (x, y, s) :
  good = []; xm, ym = [1 for i in x], [1 for i in y]; xp, yp = [0 for i in x], [0 for i in y]
  s.sort (key = lambda x : -(len (x [0])))
  for i,j in s :
    X, Y, L = j [0], j [1], len (i)
    if sum (xm [X-L:X]) and sum (ym [Y-L:Y]) :
      xm [X-L:X], ym [Y-L:Y] = xp [X-L:X], yp [Y-L:Y]; good.append ([i,j])
  return good
def rank (s) :
  s.sort (key = lambda x : x [-1][0]); for i in range (len (s)) : s [i][-1][0] = i+1
  s.sort (key = lambda x : x [-1][-1]); for i in range (len (s)) : s [i][-1][-1] = i+1
  return dict ([(tuple (j),i) for i,j in s])
def score (ss, s, x, y) :
  A, S0, S1, S2 = (len (x) * len (y)) ** 0.5, [0.], 0., 0.
  for i in ss :
    S0.append (sum ([len (s [x]) for x in i]))
    for j in range (len (i)) : S1 += len (s [i [j]]) ** 2
    for j in range (len (i)-1) : S2 += len (s [i [j]]) * len (s [i [j+1]])
  return max (S0)/A, (S1**0.5)/A, (S2**0.5)/A, ((S1+S2)**0.5)/A
def table (lx, ly) : return [[[] for j in range (ly+1)] for i in range (lx+1)]
def prod (lx, ly) : return [(i,j) for j in range (1,ly+1) for i in range (1,lx+1)]
def dcs (x, y) :
  if not x or not y : return 0., 0., 0., 0.,
  t, p = table (len (x), len (y)), prod (len (x), len (y))  # 1st cs
  for (i,j) in p :
    if x [i-1] == y [j-1] : t [i][j], t [i-1][j-1] = t [i-1][j-1]+[x [i-1]], []
  s = rank (nosub (x, y, filter (lambda x : x [0], [[t [i][j], [i,j]] for (i,j) in p])))
  t, p = table (len (s), len (s)), prod (len (s), len (s))  # 2nd cs
  for i in s : t [i [0]][i [1]] = [i]
  for (i,j) in p :
    if t [i][j] : t [i][j], t [i-1][j-1] = t [i-1][j-1]+t [i][j], []
  return score (filter (lambda x : x, [t [i][j] for (i,j) in p]), s, x, y)
```

Table 1: `Python` implementation of `dcs`. Two parameters of `dcs ()` are the two strings under comparison. Four scores will be returned by `dcs ()`, in the order of `cs0`, `cs1`, `cs2`, and `dcs` as illustrated in Fig. 1.

The numerical results are listed from Tables 2 to 5. Generally, in tasks requiring heavy reordering as English-to-Japanese and Japanese-to-English translation (Tables 2 and 4), `cs2` has the best performance in most cases, even better than RIBES. An interesting fact is that, on the Korean-to-Japanese translation (right at Table 5), which is a task nearly requiring no reordering, `cs2` is the only measure gives a moderately positive performance. It seems `cs2` is quite suitable for translation tasks with *extreme operations*, no matter heavy or none, on word reordering. However, on tasks requiring moderate reordering such as Chinese-to-Japanese and Japanese-to-Chinese translation, `dcs` gives a more stable performance. Notice `LCS`-based `cs0` is not bad a measure considering the simplicity, and `cs1` itself is not so good a measure in most cases because it does not contain much structured information.

The correlation between human-evaluation and automatic metrics may be affected by various factors. It is obvious that the **ASPEC-16** in Tables 2 and 4 has very low $\rho$'s. We consider it is because most teams switched to NMT approaches from this year. In Table 3, the **ASPEC** and **JPC** tasks show different tendencies among metrics, where it seems the **ASPEC** task requires more word reordering than that in **JPC** task. As mentioned, the automatic evaluation itself is a non-trivial task, we consider the `dcs` score (and the `cs2` in it) provides an alternative method which is intuitive and efficient enough.

# 5 Conclusion

We proposed an MT evaluation measure of `dcs` score. From the evaluation on WAT tasks, `dcs` shows comparable performances as BLEU and its `cs2` component is better than RIBES. We plan to investigate the feasibility of the method in future WAT tasks.

# References


[1] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. of ACL*, pp. 311–318, 2002.

[2] H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada, "Automatic evaluation of translation quality for distant language pairs," in *Proc. of EMNLP*, pp. 944–952, 2010.

[3] T. Nakazawa, H. Mino, I. Goto, S. Kurohashi, and E. Sumita, "Overview of the 1st workshop on Asian translation," in *Proc. of WAT*, pp. 1–19, 2014.

[4] T. Nakazawa, H. Mino, I. Goto, G. Neubig, S. Kurohashi, and E. Sumita, "Overview of the 2nd workshop on Asian translation," in *Proc. of WAT*, pp. 1–28, 2015.

[5] T. Nakazawa, H. Mino, C. Ding, I. Goto, G. Neubig, S. Kurohashi, and E. Sumita, "Overview of the 3rd workshop on Asian translation," in *Proc. of WAT*, pp. 1–46, 2016.

[6] T. Nakazawa, S. Higashiyama, C. Ding, H. Mino, I. Goto, H. Kazawa, Y. Oda, G. Neubig, and

| ASPEC-14 | $\rho$ | ASPEC-15 | $\rho$ | ASPEC-16 | $\rho$ | ASPEC-17 | $\rho$ | JPC-16 | $\rho$ |
|---|---|---|---|---|---|---|---|---|---|
| cs2 | .90 | cs2 | .95 | AMFM | .48 | BLEU-char-8 | .97 | cs2 | .77 |
| RIBES-juman | .87 | RIBES-mecab | .95 | BLEU-char-4 | .33 | BLEU-char-4 | .96 | RIBES-mecab | .67 |
| RIBES-mecab | .86 | RIBES-kytea | .95 | cs2 | .32 | BLEU-kytea | .95 | RIBES-kytea | .67 |
| cs0 | .85 | RIBES-juman | .95 | BLEU-char-8 | .26 | RIBES-mecab | .94 | RIBES-juman | .66 |
| RIBES-kytea | .85 | cs0 | .89 | cs0 | .20 | RIBES-juman | .94 | cs0 | .61 |
| dcs | .71 | dcs | .72 | dcs | .15 | cs1 | .94 | BLEU-kytea | .61 |
| BLEU-char-8 | .71 | BLEU-char-8 | .69 | BLEU-juman | .14 | cs0 | .94 | BLEU-char-4 | .60 |
| cs1 | .66 | cs1 | .64 | cs1 | .12 | dcs | .93 | BLEU-juman | .58 |
| BLEU-char-4 | .65 | BLEU-kytea | .62 | BLEU-mecab | .12 | BLEU-mecab | .93 | AMFM | .58 |
| BLEU-mecab | .64 | BLEU-juman | .62 | RIBES-juman | .10 | BLEU-juman | .93 | BLEU-mecab | .57 |
| BLEU-kytea | .64 | BLEU-mecab | .61 | BLEU-kytea | .10 | RIBES-kytea | .92 | BLEU-char-8 | .54 |
| BLEU-juman | .64 | BLEU-char-4 | .55 | RIBES-mecab | .09 | cs2 | .87 | dcs | .48 |
| AMFM | .63 | AMFM | .19 | RIBES-kytea | .08 | AMFM | .69 | cs1 | .42 |

Table 2: Pearson's $\rho$ on English-to-Japanese tasks.

| ASPEC-14 | $\rho$ | ASPEC-15 | $\rho$ | ASPEC-16 | $\rho$ | JPC-15 | $\rho$ | JPC-16 | $\rho$ |
|---|---|---|---|---|---|---|---|---|---|
| cs0 | .94 | cs2 | .94 | cs2 | .96 | cs1 | .93 | cs1 | .98 |
| cs2 | .93 | cs0 | .94 | RIBES-kytea | .96 | dcs | .93 | dcs | .98 |
| dcs | .90 | BLEU-kytea | .94 | RIBES-mecab | .95 | BLEU-char-8 | .93 | RIBES-mecab | .98 |
| BLEU-kytea | .90 | dcs | .93 | RIBES-juman | .95 | BLEU-kytea | .93 | RIBES-kytea | .98 |
| cs1 | .89 | cs1 | .93 | cs0 | .94 | BLEU-juman | .93 | BLEU-mecab | .98 |
| BLEU-char-8 | .89 | BLEU-char-8 | .93 | dcs | .91 | cs0 | .92 | BLEU-kytea | .98 |
| BLEU-mecab | .89 | BLEU-mecab | .93 | BLEU-kytea | .91 | BLEU-char-4 | .92 | BLEU-juman | .98 |
| BLEU-juman | .89 | BLEU-juman | .93 | BLEU-char-4 | .90 | BLEU-mecab | .92 | RIBES-juman | .97 |
| BLEU-char-4 | .87 | BLEU-char-4 | .92 | BLEU-mecab | .90 | RIBES-mecab | .91 | cs0 | .96 |
| RIBES-mecab | .85 | AMFM | .91 | BLEU-juman | .90 | RIBES-kytea | .91 | BLEU-char-8 | .96 |
| RIBES-kytea | .85 | RIBES-kytea | .88 | cs1 | .89 | RIBES-juman | .91 | cs2 | .94 |
| RIBES-juman | .84 | RIBES-mecab | .87 | BLEU-char-8 | .89 | cs2 | .89 | BLEU-char-4 | .93 |
| AMFM | .80 | RIBES-juman | .86 | AMFM | .88 | AMFM | .89 | AMFM | .93 |

Table 3: Pearson's $\rho$ on Chinese-to-Japanese tasks.

| 14 | $\rho$ | 15 | $\rho$ | 16 | $\rho$ | 17 | $\rho$ |
|---|---|---|---|---|---|---|---|
| cs2 | .90 | cs2 | .90 | cs2 | .35 | cs2 | .97 |
| cs0 | .89 | RIBES | .88 | cs0 | .35 | cs0 | .97 |
| RIBES | .88 | cs0 | .88 | RIBES | .23 | BLEU | .97 |
| dcs | .75 | dcs | .81 | dcs | .19 | dcs | .96 |
| BLEU | .68 | BLEU | .79 | cs1 | .13 | cs1 | .95 |
| cs1 | .63 | cs1 | .74 | BLEU | .07 | RIBES | .93 |
| AMFM | .28 | AMFM | .58 | AMFM | – | AMFM | .29 |

Table 4: Pearson's $\rho$ on Japanese-to-English tasks. (**ASPEC**, "–" means $\rho < 0$)

| ASPEC-j2z-14 | $\rho$ | JPC-k2j-15 | $\rho$ |
|---|---|---|---|
| BLEU-char-8 | .75 | cs2 | .51 |
| BLEU-char-4 | .74 | cs0 | – |
| AMFM | .74 | RIBES-kytea | – |
| cs1 | .73 | RIBES-mecab | – |
| dcs | .71 | RIBES-juman | – |
| BLEU-stdpku | .71 | AMFM | – |
| BLEU-stdctb | .71 | BLEU-char-4 | – |
| BLEU-kytea | .70 | BLEU-char-8 | – |
| RIBES-stdpku | .63 | BLEU-kytea | – |
| RIBES-stdctb | .62 | BLEU-mecab | – |
| RIBES-kytea | .62 | BLEU-juman | – |
| cs0 | .61 | dcs | – |
| cs2 | .57 | cs1 | – |

Table 5: Pearson's $\rho$ on Japanese-to-Chinese (left) and Korean-to-Japanese (right) tasks. ("–" means $\rho < 0$)


S. Kurohashi, "Overview of the 4th workshop on Asian translation," in *Proc. of WAT*, pp. 1–54, 2017.

[7] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proc. of AMTA*, pp. 223–231, 2006.

[8] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. of ACL workshop: Text summarization branches out*, 2004.

[9] C.-Y. Lin and F. J. Och, "Orange: A method for evaluating automatic evaluation metrics for machine translation," in *Proc. of COLING*, 2004.

[10] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proc of ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.

[11] A. Lavie and A. Agarwal, "METEOR: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proc. of WMT*, pp. 228–231, 2007.

[12] R. E. Banchs, L. F. D'Haro, and H. Li, "Adequacy-fluency metrics: Evaluating MT in the continuous space model framework," *IEEE TASLP*, vol. 23, no. 3, pp. 472–482, 2015.