ROUGE Working Note

Chin-Yew Lin
Information Sciences Institute
University of Southern California
cyl@isi.edu

1 ROUGE – An Automatic Evaluation Package for Summarization

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is a method to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. The measure is computed by counting the number of overlapping words between the computer-generated summary to be evaluated and the ideal summaries created by humans. This note briefly introduces three different ROUGE measures: ROUGE-N, ROUGE-L, and ROUGE-W. They will be used in the Document Understanding Conference (DUC) 2004 (http://duc.nist.gov), a large-scale summarization evaluation sponsored by NIST.

2 ROUGE-N: N-gram Co-Occurrence Statistics

Formally, ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries. ROUGE-N is computed as follows:

$$\text{ROUGE-N} = \frac{\sum_{S \in \{Referemce\ Summaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{Reference\ Summaries\}} \sum_{gram_n \in S} Count(gram_n)} \quad (1)$$

Where n stands for the length of the n-gram, $gram_n$, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries.

It is clear that ROUGE-N is a recall-related measure because the denominator of the equation is the total sum of the number of n-grams occurring at the reference summary side. A closely related measure, BLEU, used in automatic evaluation of machine translation, is a precision-based measure. BLEU measures how well a candidate translation matches a set of reference translations by counting the percentage of n-grams in the candidate translation overlapping with the references. Please see Papineni et al. (2001) for details about BLEU.

Note that the number of n-grams in the denominator of the ROUGE-N formula increases as we add more references. This is intuitive and reasonable because there might exist multiple good summaries. Every time we add a reference into the pool, we expand the space of alternative summaries. By controlling what types of references we add to the reference pool, we can design evaluations that focus on different aspects of summarization. Also note that the numerator sums over all reference summaries. This effectively gives more weight to matching n-grams occurring in multiple references. Therefore a candidate summary that contains words shared by more references is favored by the ROUGE-N measure. This is again very intuitive and reasonable because we normally prefer a candidate summary that is more similar to consensus among reference summaries.

Figure 1 shows how ROUGE-N is computed for an example from Document Understanding Conference 2003. C1 is a candidate sentence. R1 and R2 are two different references. There are 20 unigram, 19 bigram, 18 trigram, and 17 4-gram tokens from R1 and R2 which are listed in the Total column. The Match column is the sum of matches from each reference. The final score is the ratio of Match over Total, i.e. a recall score. Note that "*schizophrenic*" in C1 would match "*schizophrenics*" in R2 in a less strict matching criterion. For example, we can use a stemmer before computing the matches.

C1: *pulses may ease schizophrenic voices*
R1: *magnetic pulse series sent through brain may ease schizophrenic voices*
R2: *yale finds magnetic stimulation some relief to schizophrenics imaginary voices*

	R1	R2	Match	Total	Score
ROUGE$_1$	*may, ease, schizophrenic, voices*	*voices*	5	20	0.2500
ROUGE$_2$	*may ease, ease schizophrenic, schizophrenic voices*	NA	3	19	0.1669
ROUGE$_3$	*may ease schizophrenic, ease schizophrenic voices*	NA	2	18	0.1111
ROUGE$_4$	*may ease schizophrenic voices*	NA	1	17	0.0588

NA means no matches.

Figure 1. An example shows how ROUGE$_n$ is computed.

An initial study (Lin and Hovy 2003) indicates that automatic evaluation using the unigram version of ROUGE-N, i.e. ROUGE-1, correlates well with human evaluations based on various statistical metrics.

2.1 Multiple References

So far, we only demonstrated how to compute ROUGE-N using a single reference. When multiple references are used, we compute pairwise summary-level ROUGE-N between a candidate summary s and every reference, r_i, in the reference set. We then take the maximum of pairwise summary-level ROUGE-N scores as the final multiple reference ROUGE-N score. This can be written as follows:

$$ROUGE\text{-}N_{multi} = \text{argmax}_i\, ROUGE\text{-}N(r_i, s) \quad (2)$$

This procedure is also applied to computation of ROUGE-L (Section 3) and ROUGE-W (Section 4)
In the implementation, we use a Jackknifing procedure. Given N references, we compute the best score over N sets of N-1 references. The final ROUGE-N score is the average of the N ROUGE-N scores using different N-1 references. The Jackknifing procedure is adopted since we often need to compare system and human performance and the reference summaries are usually the only human summaries available. Using this procedure, we are able to estimate average human performance by averaging N ROUGE-N scores of one reference vs. the rest N-1 references. Although the Jackknifing procedure is not necessary when we just want to compute ROUGE scores using multiple references, it is applied in all ROUGE score computations in the ROUGE evaluation package.
In the next section, we describe a ROUGE measure based on longest common subsequences between two summaries.

3 ROUGE-L: Longest Common Subsequence

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 $x_{ij} = z_j$ (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 in identifying cognate candidates during construction of N-best translation lexicon 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 LCS as an approximate string matching algorithm. Saggion et al. (2002) used normalized pairwise LCS to compare similarity between two texts in automatic summarization evaluation.
However, they did not provide any correlation analysis with human judgments. Normalized pairwise LCS as described in Saggion et al. (2002) is available through MEAD:

http://www.summarization.com/mead

3.1 Sentence-Level LCS

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

$$R_{lcs} = \frac{LCS(X,Y)}{m} \quad (2)$$

$$P_{lcs} = \frac{LCS(X,Y)}{n} \quad (3)$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \quad (4)$$

Where $LCS(X,Y)$ is the length of a longest common subsequence of X and Y, and β is the ratio between P_{lcs} and R_{lcs}, i.e. $\beta = P_{lcs}/R_{lcs}$. In DUC, β is set to 0. Therefore, only R_{lcs} is considered. We call the LCS-based F-measure, i.e. Equation 4, ROUGE-L. Notice that ROUGE-L is 1 when $X = Y$ since $LCS(X,Y) = min(m, n)$; 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. We adopt F-measure for the following reasons:

(1) To avoid arbitrary length adjustment functions such as the brevity penalty used in BLEU;
(2) It is well known in the information retrieval and natural language processing communities;
(3) It is an intuitive way to adjust the interaction between recall and precision through the β parameter;
(4) It has nice theoretical properties as a composite function.

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 4 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 summary sentence) that are also present in Y (candidate summary sentence); 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 the following example:

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

We only consider ROUGE-2, i.e. N=2, for the purpose of explanation. Using S1 as the reference and S2 and S3 as the candidate summary sentences, S2 and S3 would have the same ROUGE-2 score, since they both have one bigram, i.e. "the gunman". However, S2 and S3 have very different meanings. In the case of ROUGE-L, S2 has a score of 3/4 = 0.75 and S3 has a 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 suffers one disadvantage that it only counts the main in-sequence words; therefore, other alternative LCSes and shorter sequences are not reflected in the final score. For example, given 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. ROUGE-2 would prefer S4 than S3.

3.2 Summary-Level LCS

Previous section described how to compute sentence-level LCS-based F-measure score. When applying to summary-level, we take the union LCS matches between a reference summary sentence, r_i, and every candidate summary sentence, c_j. Given a reference summary of u sentences containing a total of m words and a condidate summary of v sentences containing a total of n word, the summary-level LCS-based F-measure can be computed as follows:

$$R_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i, C)}{m} \quad (5)$$

$$P_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i, C)}{n} \quad (6)$$

$$F_{lcs} = \frac{(1+\beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \quad (7)$$

Again $\beta = P_{lcs}/R_{lcs}$ and is set to 0 in DUC. $LCS_{\cup}(r_i, C)$ is the union LCS score of reference sentence r_i and candidate summary C. For example, if $r_i = w_1\ w_2\ w_3\ w_4\ w_5$, and C contains two sentences: $c_1 = w_1\ w_2\ w_6\ w_7\ w_8$ and $c_2 = w_1\ w_3\ w_8\ w_9\ w_5$, then LCS of r_i and c_1 is "$w_1\ w_2$" and LCS of r_i and c_2 is "$w_1\ w_3\ w_5$". The union LCS of r_i, c_1, and c_2 is "$w_1\ w_2\ w_3\ w_5$" and $LCS_{\cup}(r_i, C) = 4/5$.

3.3 ROUGE-L vs. Normalized Pairwise LCS

The normalized pairwise LCS proposed by Radev et al. (page 51, 2002) between two summaries S1 and S2 is written as follows:

$$LCS(S_1, S_2)_{MEAD} = \frac{\sum_{s_i \in S_1} \max_{s_j \in S_2} LCS(s_i, s_j) + \sum_{s_j \in S_2} \max_{s_i \in S_1} LCS(s_i, s_j)}{\sum_{s_i \in S_1} length(s_i) + \sum_{s_j \in S_2} length(s_j)} \quad (8)$$

Assuming S1 has m words and S2 has n words, Equation 8 can be rewritten as Equation 9 due to symmetry:

$$LCS(S_1, S_2)_{MEAD} = \frac{2 * \sum_{s_i \in S_1} \max_{s_j \in S_2} LCS(s_i, s_j)}{m+n} \quad (9)$$

We then define MEAD LCS recall ($R_{lcs\text{-}MEAD}$) and MEAD LCS precision ($P_{lcs\text{-}MEAD}$) as follows:

$$R_{lcs\text{-}MEAD} = \frac{\sum_{s_i \in S_1} \max_{s_j \in S_2} LCS(s_i, s_j)}{m} \quad (10)$$

$$P_{lcs\text{-}MEAD} = \frac{\sum_{s_i \in S_1} \max_{s_j \in S_2} LCS(s_i, s_j)}{n} \quad (11)$$

We can rewrite Equation (9) in terms of $R_{lcs\text{-}MEAD}$ and $P_{lcs\text{-}MEAD}$ with a constant parameter $\beta = 1$ as follows:

$$LCS(S_1, S_2)_{MEAD} = \frac{(1+\beta^2) R_{lcs-MEAD} P_{lcs-MEAD}}{R_{lcs-MEAD} + \beta^2 P_{lcs-MEAD}} \quad (12)$$

Equation 12 shows that normalized pairwise LCS as defined in Radev et al. (2002) and implemented in MEAD is also a F-measure with $\beta = 1$. Sentence-level normalized pairwise LCS is the same as ROUGE-L with $\beta = 1$. Besides setting $\beta = 1$, summary-level normalized pairwise LCS is different from ROUGE-L in

how a sentence gets its LCS score from its references. Normalized pairwise LCS takes the best LCS score while ROUGE-L takes the union LCS score.

4 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 recurrent relations can be written as follows:

(1) If $x_i = y_j$ Then
    // *the length of consecutive matches at*
    // *position i-1 and j-1*
    $k = w(i\text{-}1,j\text{-}1)$
    $c(i,j) = c(i\text{-}1,j\text{-}1) + f(k+1) – f(k)$
    // *remember the length of consecutive matches*
    // *at position i, j*
    $w(i,j) = k+1$
(2) Otherwise
    If $c(i\text{-}1,j) > c(i,j\text{-}1)$ Then
        $c(i,j) = c(i\text{-}1,j)$
        $w(i,j) = 0$ // *no match at i, j*
    Else $c(i,j) = c(i,j\text{-}1)$
        $w(i,j) = 0$ // *no match at i, j*
(3) $WLCS(X,Y) = c(m,n)$

Where c is the dynamic programming table, $0 <= i <= m$, $0 <= j <= n$, 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:

$$R_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(m)}\right) \quad (13)$$

$$P_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(n)}\right) \quad (14)$$

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

Where f^{-1} is the inverse function of f. In DUC, $\beta = P_{wlcs}/R_{wlcs}$ is set to 0. Therefore, only R_{wlcs} is considered. We call the WLCS-based F-measure, i.e. Equation 15, ROUGE-W. Using Equation 15 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. Users can specify the weighting factor α through the command line option -*w*.

5 ROUGE Evaluation Package

ROUGE is available free for research purpose. To download it and obtain more details about how ROUGE works, please go to http://www.isi.edu/~cyl/ROUGE.

References

Cormen, T. R., C. E. Leiserson, and R. L. Rivest. 1989. *Introduction to Algorithms*. The MIT Press.

Davison, A. C. and D. V. Hinkley. 1997. *Bootstrap Methods and Their Application*. Cambridge University Press.

Lin, Chin-Yew and E.H. Hovy 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *Proceedings of 2003 Language Technology Conference* (HLT-NAACL 2003), Edmonton, Canada.

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.

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)*.

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.

Dragomir Radev, S. Teufel, H. Saggion, W. Lam, J. Blitzer, A. Gelebi, H. Qi, E. Drabek, and D. Liu. 2002. Evaluation of Text Summarization in a Cross-Lingual Information Retrieval Framework. Technical report, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA.

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