Review Article

A Review of the Research on the Evaluation Metrics for Automatic Grammatical Error Correction System

Manli Long, Yan Wang, Yifei Peng, and Wanwu Huang

School of Foreign Languages, Hubei University of Technology, Wuhan, Hubei Province 430068, China

Correspondence should be addressed to Wanwu Huang; huangwanwu@mail.hbut.edu.cn

Received 11 July 2022; Revised 1 September 2022; Accepted 12 September 2022; Published 4 October 2022

Academic Editor: Santosh Tirunagari

Copyright © 2022 Manli Long et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The evaluation of an automatic grammatical error correction system is an important content in the field of automatic grammatical error correction. This paper summarizes the technical routes of the four most representative evaluation metrics of the automatic grammatical error correction systems. Firstly, it introduces the characteristics and composition of each metric, then summarizes its defects, and finally puts forward some suggestions for the future development of the metrics. This paper holds that the application of natural language processing technology should be strengthened in the future development of evaluation metrics.

1. Introduction

Machine learning is the study of how to train computers to do tasks like speech recognition, data analysis, computer vision, and natural language processing [1]. Rule-based and machine learning-based methodologies are the two main approaches used in the design of contemporary linguistic experiments and the development of natural language processing systems. The approaches that use supervised learning, or those that are based on manually created training data for learning, get the best results in practical applications of machine learning (ML). A system built on a hybrid method should produce better outcomes, according to what can be termed a general rule for the combining of these two approaches [2].

For the creation of the grammar checking system, we use a new set of matching standards, which aims to identify the preposition usage problems of second language learners. An F-score of 40% and good precision were found in a modest study of this set of rules' performance [3].

Grammatical error correction (GEC) is an important task in the field of natural language processing (NLP) [4]. GEC research has primarily gone through three stages of development: initially, simple string matching and substitution; later, using rules for syntactic error analysis; and currently, using data-driven ways to extract features from the data, and using machine learning algorithms to build a model to detect and correct errors. The ways that based on machine learning have also experienced two stages of development, from the machine translation method based on statistics [5–7] to the most cutting-edge machine translation method based on the neural network [8–10]. The continuous development of GEC in the recent 20 years also drives the progress and improvement of its own evaluation metrics.

As automatic scoring technology for English composition has grown and developed, computer-aided composition marking systems based on this technology have started to appear in colleges and universities to help with the teaching of English writing. The system can translate the probability distribution of certain sentences or word sequences by creating a language model. The BP neural network, which is an adaptive learning model, provides additional advantages when addressing the link between complicated variables [11].

The creation of GEC systems is closely linked to the creation of its evaluation metrics. That’s because with the continuous optimization of GEC system research methods, the traditional evaluation metrics cannot meet the needs of multidimensional analysis of GEC system results.

In this context, this paper will introduce and comment on the four most representative automatic grammatical error correction evaluation metrics: Max Match, I-measure, GLEU, and ERRANT.
2. Method

The following section describes method of current work.

2.1. Evaluation Metrics of GEC System. The incorrect and wrong sentence that was originally written is typically referred to as the source sentence in the evaluation index of grammatical error correction systems. The sentence that was manually marked and corrected is known as the reference sentence. The higher the coincidence degree, the better the result and the better the system performance.

The method of evaluating edit distance includes three measurement dimensions: \( P \) (precision), \( R \) (recall), and \( F \). \( P \) is used to calculate the precision rate of error correction results, and \( R \) is used to calculate the recall rate, as shown in formulas (1) and (2).

\[
P = \frac{\sum_{i=1}^{n} |e_i \cap g_i|}{\sum_{i=1}^{n} |e_i|},
\]

\[
R = \frac{\sum_{i=1}^{n} |e_i \cap g_i|}{\sum_{i=1}^{n} |g_i|}.
\]

In the formula, \( e_i \) represents the editing set between the source sentence and the hypothetical sentence \( I \); \( g_i \) is the optimal editing set between the source sentence and the reference sentence \( I \); \( |e_i \cap g_i| \) is the intersection of edits between \( e_i \) and \( g_i \); \( |e_i| \) is the number of edits in \( e_i \); and \( |g_i| \) is the number of edits in \( g_i \). The value of the intersection of \( e_i \) and \( g_i \) is shown in formula (3):

\[
e_i \cap g_i = \{ e \in e_i \mid \exists g \in g_i \ (\text{match}(e, g)) \}.
\]

Taking the sentence \( \text{Our baseline system feed into PB-SMT pipeline} \) as an example, if the manually modified set \( g = \{ \text{feed} \rightarrow \text{feeds}, \text{feed} \rightarrow \text{feed a word} \} \), the modified set \( e = \{ \text{feed} \rightarrow \text{feeds} \} \), \( |e_i \cap g_i| \) is 1, then the precision of the system is 100% and the recall rate is 50%.

In an ideal evaluation, both \( P \) and \( R \) values are expected to be as high as possible. But in practice, the precision rate and the recall rate are inconsistent under certain circumstances. Suppose that the optimal editing set \( g = 10 \), the modified set \( e_1 = 5 \), \( |e_i \cap g_i| = 3 \) in system 1 and the modified set \( e_2 = 8 \), \( |e_i \cap g_i| = 4 \) of system 2. Under these conditions, the \( P \) value of system 1 is 0.6, \( R \) value is 0.3, and system 2 has a \( P \) value of 0.5 and an \( R \) value of 0.4. The results show that the \( P \) value of system 1 is higher than that of system 2, but the \( R \) value is lower than that of system 2. Therefore, for the convenience of comparison, the \( F \) (F-measure or F-score) [14] is introduced to evaluate the \( P \) value and the \( R \) value as a whole. The \( F \) value is the weighted harmonic average of the \( P \) and \( R \) values, as shown in formula (4):

\[
F_a = \frac{(\alpha^2 + 1) \times P \times R}{\alpha^2 P + R}.
\]  

In the \( M^2 \) evaluation, the value of \( \alpha \) is usually set to 1, and the calculation formula of \( F_1 \) is shown in formula (5).

\[
F_1 = 2 \times \frac{P \times R}{P + R}.
\]

Later, CoNLL-2014 shared mission [15] was introduced and \( M^2 \) was used as the official evaluation indicator for the mission evaluation. But because of the nature of the data set, the researchers changed the value of weight \( \alpha \) from 1 to 0.5. Because in the shared task evaluation process, the task recognizes that the precision rate weighs more than the recall rate and thus gives it a higher weight, twice as much weight as before, and twice as much precision, \( F_{0.5} \) as shown in formula (6).

\[
F_{0.5} = \frac{(1 + 0.5^2) \times P \times R}{0.5^2 \times P + R}.
\]

Currently, \( M^2 \) is the evaluation index most frequently used in the field of GEC, largely as a result of CoNLL-2014’s shared task’s promotion. This tool has evolved into the standard for figuring out the GEC system’s precision and recall rate. There is a strong association between this evaluation metric and manual error correction [16].

2.2. MaxMatch. Max match, also known as \( M^2 \) [12], mainly evaluates the error correction effect of GEC system based on the phrase level edit lattice. The metric first calculates the editing lattice between the source sentence and the hypothetical sentence. The Levenshtein distance [13] is based on many inserts, deletions, and replacements needed to change one string into another, is used as the basis for the calculation process. For example, to convert \( \text{kitten} \) into \( \text{sitting} \), the conversion steps are as follows: \( \text{kitten} (k \rightarrow s) \), \( \text{sitting} (e \rightarrow g) \), and the editing distance is 3. The above example is the calculation method of editing distance at the character level, while \( M^2 \) is mainly based on the calculation at the phrase level, that is, the minimum editing operation required to replace one phrase with another. \( M^2 \) matches the editing result of the error correction system with the editing between the source sentence and the reference sentence. The higher the coincidence degree, the better the result and the better the system performance.

2.3. F-Measure. The construction system of \( M^2 \), the official evaluation metric of the CoNLL-2014 shared tasks, has received high recognition but is still limited. Felice and Briscoe [17] believe that (1) relying only on the output results cannot define the difference between the baseline(unmodified system for comparison) system that “does nothing” and other systems that only propose wrong corrections, because their \( F \) values are all 0; (2) when multiple correction annotations are used on sentences, the performance of the system is underestimated because the metric automatically selects the maximum \( F \) value instead of mixing all the corrections to calculate the
output. As shown in Table 1, the error correction system provides two change schemes, but the system only chooses the combined output with the largest value of F; (3) partial matches are ignored in the evaluation, as shown in Table 2, where the system results are different from the reference sentence results but have an output, but the F value is 0; and (4) editing at the phrase level produces misleading results that often do not reflect effective improvements. As shown in Table 3, the first predicted improvement is significantly better than the second, but the F value is lower:

In light of the above shortcomings, Felice and Briscoe propose a new metric, I-measure. First of all, in view of the problem that \( M^2 \) underestimates the performance of GEC system under multiple annotations, a new annotation method is proposed, which provides different modification schemes based on the same error type. For example, the source sentence “This machine is designed for help people.” Firstly, the error types are defined. There are two kinds of errors in this sentence: SVA (subject-predicate agreement) and V-form (verb form). Then the different modification schemes are annotated under the error types as shown in Table 4. All the alternatives are mutually exclusive. It is because of mutual exclusion that the system can directly combine them to form all kinds of valid and correct sentences, and the problem of multiple annotations can be solved effectively. Secondly, by three-way alignments (source sentence, hypothetical sentence, and reference sentence), the matching of error detection and correction is calculated. After finding the best alignment of source sentence, hypothetical sentence, and reference sentence, the system will be evaluated.

Each word aligned after three-way alignment is classified as TP (true positive), TN (true negative), FP (false positive), and FN (false negative) under the WAS evaluation system [18] which is used in this study. Given that aligned words are represented as \( w^{src} \) (word in source sentence), \( w^{hyp} \) (word in hypothetical sentence), and \( w^{ref} \) (word in reference sentence) under source sentence, hypothesis sentence, and reference sentence, respectively, TP, TN, FP, and FN classifications are defined as follows:

\[
\begin{align*}
TP: w^{src} \neq w^{ref} \text{ and } w^{hyp} = w^{ref} \\
TN: w^{src} \neq w^{hyp} = w^{ref} \\
FP: w^{src} \neq w^{hyp} \text{ and } w^{hyp} \neq w^{ref} \\
FN: w^{src} \neq w^{ref} \text{ and } w^{hyp} \neq w^{ref}
\end{align*}
\]

Accuracy gives equal weight to all indicators, but when the sum of different TP and TN is the same, the indicators will not be able to distinguish the advantages and disadvantages of the system. Therefore, the weight is introduced, and the calculation formula is shown in (8):

\[
\begin{align*}
W\text{Acc} &= \frac{w \cdot TP + TN}{w \cdot TP + TN + w \cdot (FP - FPN/2) + (FN - FPN/2)} \\
&= \frac{w \cdot TP + TN}{w \cdot TP + TN + w \cdot FP - w \cdot FPN/2 + FN - FPN/2} \\
&= \frac{w \cdot TP + TN}{w \cdot (TP + FP) + TN + FN - (w + 1) \cdot FPN/2}.
\end{align*}
\]

By comparing the accuracy rate of the assessed system with the accuracy rate obtained through the baseline system, as indicated in formula (9), the I value which is the system's final score is obtained after getting the weighted accuracy rate.

\[
I = \begin{cases} 
\frac{[W\text{Accsys}]}{W\text{Accbase}}, & \text{if } W\text{Accsys} = W\text{Accbase}, \\
\frac{W\text{Accsys} - W\text{Accbase}}{1 - W\text{Accbase}}, & \text{if } W\text{Accsys} > W\text{Accbase}, \\
W\text{Accsys} - 1, & \text{otherwise.}
\end{cases}
\]

2.4. GLEU. A variant of the machine translation system measure BLEU (bilateral evaluation understudy) [19], called GLEU (generalized language evaluation understanding), was put forth by IBM researchers in 2001. GLEU is largely used to assess the output of machine translation models, which is typically between 0.0 and 1.0. If the two sentences match perfectly, BLEU = 1.0. Conversely, if the two sentences mismatch perfectly, BLEU = 0.0 [20]. The core of BLEU translation evaluation metric is to detect the number of co-occurrence words between the hypothetical sentences and the reference sentences. The specific implementation method is to calculate the n-grams of the hypothetical sentence and the reference sentence and then count the number of matches to get the score. The more grams the system translation matches with the manual reference translation, the higher the BLEU score. Examples are as follows [21]:

Reference: this is a small test.
Candidate: this is a test.

Table 5 shows that the BLEU score under 1-gram is 0.8 since the hypothetical sentence shares 4 words with the reference sentence. The number of the words in the hypothetical sentence divided by the words in the reference sentence is the final score.

Under 2-gram, as shown in Table 6:

Every two words in the sentence are divided into a 2-gram group. The calculation logic is the same as that of 1-
Under the conditions of the same reference sentence and hypothetical sentence, BLEU score is 0.5. Napoles et al. [22] contend that because there is still a crucial distinction between translation tasks and error correction tasks, it is inaccurate to consider machine translation as merely a monolingual translation. Direct application of BLEU to GEC tasks could result in less-than-ideal output scores. Because of this, researchers have created a simple BLEU metric variant called GLEU that is suited to the requirements of the error correction task. The accuracy of the GEC system is calculated through the comparison of reference sentence and source sentence, giving more weight to the gram with correct correction, rewarding the correct correction result and the correct source text without correction, and punishing the gram with incorrect correction.

The calculation formula is as follows (10):

$$\text{GLEU} = \text{BP} \cdot \exp \left( \sum_{n=1}^{N} \ln n! \log p_n \right),$$

(10)

$$p_n = \frac{N(H, R) - (N(H, S) - N(H, S, R))}{N(H)},$$

(11)

$$\text{BP} = \begin{cases} 1 & \text{if } h > r \\ \exp \left( 1 - \frac{r}{h} \right) & \text{if } h \leq r \end{cases},$$

(12)

In the formulas, $p_n$ is the accuracy after $n$-gram calculation and $\text{BP}$ is the penalty factor. In formula (11), $H$ is the length of the hypothetical sentence and $R$ is the length of the reference sentence. The function of penalty factor is to avoid the bias of system scoring. In the scoring process, the matching degree of $n$-gram may become better with the
shortening of sentence length. Therefore, in order to control this situation, the length of the sentence will be taken into account in the calculation. When the length of the hypothetical sentence is greater than the source sentence, the penalty factor is 1 and no penalty will be imposed. When the length of the hypothetical sentence is greater than the source sentence, the punishment will be carried out.

\[ N \] is the value of \( n \) in \( n \)-gram of GLEU formula, and its upper limit is 4. \( N(H, R) \) is the overlapping \( n \)-grams in the hypothetical sentence and the source sentence, and \( N(H, S, R) \) is the overlapping \( n \)-grams in the hypothetical sentence, the source sentence, and the reference sentence, respectively. \( w_n \) is the weighted average adopted by the system, and the value is \( 1/N \).

2.5. ERRANT. ERRANT (ERRor ANnotation Toolkit) [23] was proposed by ALTA Research Institute of Cambridge University. The evaluation metric is mainly divided into two steps. Firstly, the phrase level editing between hypothetical sentence and source sentence pair is extracted, and then the editing is classified according to the error type based on rules. The extraction of phrase level editing mainly adopts a new method proposed by Felice [24]—edit extraction using a linguistically enhanced alignment algorithm supported by a set of merging rules. In terms of rule classification, ERRANT first applies the part-of-speech (POS) tagging techniques already in use to identify and categorize errors at the part-of-speech level and then adds rules to pinpoint errors like missing, redundant, and replacement. It finally broadens the approach to locate errors outside the part-of-speech, like spelling, word placement, and so forth. The final coding is about 50 rules. The following table lists the main 25 rules and their examples and comments in Table 7.

The calculation method of \( F \) value in errant is the same as that in \( M^2 \). However, the advantage of ERRANT is that the internal classification of the method is transparent, the category of errors can be clearly identified, and the requirements for data annotation are not high. Users can clearly know the causes of wrong classification, which is more beneficial to the development of GEC system.

3. Discussion

Table 8 summarizes the calculation core, advantages, and disadvantages of the above four evaluation metrics.

From calculation point, the majority of evaluation metrics continue to compare the editing differences between the source sentence and the hypothetical sentence and the source sentence and the reference sentence in order to assess the benefits and drawbacks of the model. The main difference lies in the difference of calculation methods. \( M^2 \) and ERRANT evaluate the performance of the model by calculating the accuracy rate and recall rate of the error correction system, and \( L \)-measure calculates the quality of the output samples produced by the error correction system based on the accuracy rate. The main advantage of the method based on edits statistics is that the calculation is simple, but there is no way to obtain the information at the semantic level because all its operations are completely
Table 8: Comparison of four evaluation metrics.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Calculation core</th>
<th>Strength</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>M(^2)</td>
<td>Editing</td>
<td>1. The calculation is relatively simple and convenient</td>
<td>1. The difference between &quot;no correction&quot; and &quot;error correction&quot; cannot be captured</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Help of CoNLL-2014 shared tasks</td>
<td>2. The selective output of correction results may lead to the underestimation of system performance</td>
</tr>
<tr>
<td>I-measure</td>
<td>Editing</td>
<td>1. Multiple annotations make the systematic evaluation more objective</td>
<td>1. Heavy marking workload</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Consider a wider range of sample extraction</td>
<td>2. There is no way to obtain semantic information based on the structure of the text itself</td>
</tr>
<tr>
<td>GLEU</td>
<td>Language model</td>
<td>1. No need to make detailed comments on the location and error type of each correction, and the annotation cost is small</td>
<td>1. Regardless of meaning.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Adapt to multiple languages</td>
<td>2. Without directly consider sentence structure.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3. Cannot handle language-rich text well.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4. Inconsistent with human judgment [25]</td>
</tr>
<tr>
<td>ERRANT</td>
<td>Editing</td>
<td>1. Low requirements for database</td>
<td>1. Artificial rules have limited coverage of error types high postmaintenance cost</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Be able to conduct specific performance evaluation for different error types</td>
<td></td>
</tr>
</tbody>
</table>

calculated based on the structure of the text itself, the modification validity of the system on the semantic errors of the text cannot be effectively evaluated, and the cost of manual annotation in the early stage is high at beginning. GLEU abandons the traditional editing and extraction mode and uses the translation evaluation method to calculate the difference between the hypothetical sentence and the reference sentence. This translation-evaluation-based algorithm greatly reduces the manual intervention in the early stage of model construction.

From the perspective of individual metrics, although \(M^2\) is the most widely used evaluation metric, problems of the metrics themselves emerge in the process of continuous use.

Firstly, as a result of its calculation method, it is not possible to distinguish the extreme cases at first, and it is impossible to distinguish between the results of zero F value and those of no correction and total error correction for GEC system.

Secondly, the selected output of indicators will underestimate the performance of the system [26]. But overall, \(M^2\) is simple to use, and with the promotion of shared tasks, \(M^2\) is still the first choice for all major GEC system evaluation indicators. I-measurements are mainly based on \(M^2\), with multilevel annotations for editing calculations, more comprehensive sampling than \(M^2\), and richer scenarios to consider. GLEU jumps out of the framework of extracting editing features from the original indicators and introduces the translation evaluation model into the field of error correction, which solves the problem of complex annotations of the above two models [27–29], with less computational cost. Moreover, the evaluation form of language model has no requirement on language type and can be used in a wider range. However, in the process of calculating the result of the metric, gram calculation does not take into account the issue of semantic errors and sentence structural errors, and excessive focus is put on local errors and overall grammatical coherence is ignored. ERRANT classifies errors based on grammar rules and then evaluates them. The overall calculation method is similar to \(M^2\), but the advantage lies in the classification process. By classifying and calculating the index, we can output the performance of different systems in different error types, which is more meaningful for the further development of GEC system. The disadvantage is that even though the number of rules written is broad enough, it still cannot cover all the error types in predicting, new errors are discovered later, and rules need to be manually added to update maintenance system, which is costly.

4. Conclusion

The study presented above leads this paper to the conclusion that future evaluation criteria should be more oriented toward lowering the investment in manual labeling and labor and strengthening the use of natural language processing technologies.

Firstly, the development of the evaluation metrics of grammar error correction system partially reflects the development process of natural language processing technology—from the rule era of manual labeling errors and a large number of manual intervention to the deep learning era of manual work reduction and machine processing of corpus. However, there is still room for considerable progress. From the perspective of individual evaluation indicators, the evaluation results still have limitations. The focus of all indicators is still on the comparative evaluation of the results of local error correction, which does not reflect the evaluation of the correction effect of long difficult sentences and text level, and the scope of evaluation is very limited. In the future, the improvement of indicators should focus more on the syntactic and textual levels.

Secondly, on the whole, although the application of methods in evaluation metrics has been improved, manual labeling is still in a dominant position. It does not fully
reflect the idea of deep learning. The future methods should be more integrated into machine work and reduce the manual intervention in corpus.

**Data Availability**

The data used to support the findings of this study are available from the corresponding author upon request.

**Conflicts of Interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**References**


