NLG-METRICVERSE: An End-to-End Library for Evaluating Natural Language Generation

Giacomo Frisoni, Antonella Carbonaro, Gianluca Moro, Andrea Zammarchi and Marco Avagnano
Department of Computer Science and Engineering (DISI)
University of Bologna, Via dell’Università 50, I-47522 Cesena, Italy
{giacomo.frisoni,antonella.carbonaro,gianluca.moro}@unibo.it
{andrea.zammarchi3, marco.avagnano}@studio.unibo.it

Abstract

Driven by deep learning breakthroughs, natural language generation (NLG) models have been at the center of steady progress in the last few years, with a ubiquitous task influence. However, since our ability to generate human-indistinguishable artificial text lags behind our capacity to assess it, it is paramount to develop and apply even better automatic evaluation metrics. To facilitate researchers to judge the effectiveness of their models broadly, we introduce NLG-METRICVERSE—an end-to-end open-source library for NLG evaluation based on Python. Our framework provides a living collection of NLG metrics in a unified and easy-to-use environment, supplying tools to efficiently apply, analyze, compare, and visualize them. This includes (i) the extensive support to heterogeneous automatic metrics with n-arity management, (ii) the meta-evaluation upon individual performance, metric-metric and metric-human correlations, (iii) graphical interpretations for helping humans better gain score intuitions, (iv) formal categorization and convenient documentation to accelerate metrics understanding. NLG-METRICVERSE aims to increase the comparability and replicability of NLG research, hopefully stimulating new contributions in the area.

1 Introduction

Natural language generation (NLG) is a sub-field of natural language processing (NLP) concerned with automatically generating human-understandable text from input data, like prompts, tables, graphs, and images. Remarkably, the ability of a machine to produce text indistinguishable from that written by humans is a pre-requisite for Artificial General Intelligence (AGI)—the holy grail of AI. Recent advancements in deep learning have yielded tremendous improvements in the NLP sector, making NLG the object of fast-growing interest from the research community, as aptly demonstrated by GPT-3 (Brown et al., 2020). Pre-trained language models with transformer-based architectures (Kalyan et al., 2021) continue to push the envelope with unprecedented performance and encourage more and more applications. Indeed, today NLG includes a wide variety of tasks, such as machine translation, single/multi-document summarization, data-to-text, text-to-text, dialogue generation, free-form question answering, and image/video captioning (Gatt and Krahmer, 2018).

As NLG models get better over time, accurately evaluating them is becoming an increasingly pressing priority for tracking progress in the area and convincingly recognizing state-of-the-art systems. However, the assessment of NLG model output is notoriously a challenging problem (Howcroft...
It involves the consideration of multiple intrinsic quality dimensions (e.g., informativeness, fluency, coherence, adequacy) and open-ended scenarios, where different plausible or equal-meaning responses may exist for the same user input. Human evaluation is typically regarded as the gold standard. Nevertheless, designing crowdsourcing experiments accompanied by elaborated guidelines is an expensive and high-latency process, which does not easily fit in a daily model development pipeline with the need for automatic benchmarking and tuning at scale. Furthermore, as NLG models improve, evaluators are asked to read longer passages of text conditioned on large amounts of context. In these cases, errors are often content-based (e.g., factual inaccuracies or context inconsistencies) rather than fluency-based, making superficial reads and non-expert annotators insufficient (Clark et al., 2021).

Given these issues, NLG researchers have settled for automatic evaluation metrics computing a holistic or dimension-specific score, an acceptable proxy for effectiveness and efficiency. Unfortunately, despite the rapid surge of machine-generated language, evaluation metrics have fallen behind, leaning on the conservative use of surface-level lexical similarities, which fail to cope with diversity and capture the text's underlying meaning. To overcome this severe bottleneck, the community has witnessed—in a relatively short time—a prolific, variegated, and original research production. New NLG metrics are constantly being proposed in top conferences, exhibiting one or more of the following characteristics: (i) use of contextualized word embeddings (Zhang et al., 2020), (ii) pre-training on massive unlabeled corpora (Sellam et al., 2020), (iii) fine-tuning on data annotated with human judgments (Kaur et al., 2020), (iv) management of task-specific nuances (Dhingra et al., 2019; Wang et al., 2020).

Per contra, NLG metrics today are often designed and implemented from scratch with distinct environments, assumptions, properties, settings, benchmarks, and features. Such heterogeneity and disregregation make them difficult to compare or move to slightly different contexts. Concretely, the absence of a collective and continuously updated repository—well-documented and covering the entire NLG evaluation pipeline—discourages the use of modern solutions and slows down their understanding and practical application. Such barrier is highlighted also by the latest surveys (Sai et al., 2022). In the quest to fill this gap, we present NLG-METRICVERSE², an open-source (MIT licensed) end-to-end library for NLG evaluation, devised to provide a shared and collaborative codebase for fast application, analysis, comparison, visualization, and prototyping of automatic metrics.

The rest of the paper is organized as follows. First, we enumerate the design principles at the basis of NLG-METRICVERSE (§3), clarify the context, and summarize prior work related to this project (§2). Then, we describe the overarching NLG evaluation framework that constitutes the conceptual foundation for our contributions (§4). Next, we examine the main modules of the library: metrics, meta-evaluation, and visualization (§5). Lastly, we close the discussion and point out possible extensions (§7).

2 Background and Related Work

Early lexical NLG metrics, such as the BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), still appear to dominate the landscape, waiting for feasible, robust, and widely-adopted alternatives. Despite the high number of criticisms and studies proving their poor correlation with human judgment (Zhang et al., 2004), the popularity of first-generation metrics has not declined but expanded with the emergence of deep neural networks and new tasks. Simplicity, consistency, unsupervision, lightweight, and fast computation are the central basis of this success.

However, it has become increasingly clear that such adoption is often not prudent. Metrics measuring surface-level overlap are unsuitable for advanced evaluation, especially for modern text generation systems trained on mammoth data and with impressive paraphrasing capabilities (Mathur et al., 2020)—where ideal metrics should be sensitive to the underlying semantics. As a remedy, NLG researchers have started injecting learned/learnable components into their metrics, moving from a discrete space of word tokens to a continuous high-dimensional space of word vectors, thereby capturing distributional semantics. Over the years, many strong NLG evaluation metrics have been proposed, particularly transformer-based, like BLEURT (Sel-
lam et al., 2020), BERTScore (Zhang et al., 2020), and BARTScore (Yuan et al., 2021).

The trend towards the definition of model-based metrics and the resolution of task-specific needs have created a fertile ground for research. According to Sai et al. (2022), from 2002 (when BLEU was proposed) to 2014 (when Deep Learning became prevalent), there were only about 10 automatic NLG evaluation metrics in use; since 2015, at least 36 new metrics have appeared. On the other side, metrics are often scattered online, non-maintained, undocumented, implemented in various languages, inconsistent with the paper results. This not only hampers reproducibility but also inhibits scalability, as each research paper ends up creating its own implementation almost from scratch. Some libraries have already tried to make an integrated environment. To our best knowledge, NLGEval ( Sharma et al., 2017), HuggingFace Datasets (Lhoest et al., 2021), Evaluaute, TorchMetrics (Detlefsen et al., 2022), and Jury (Cavusoglu et al., 2022) are the only resources available. However, none of them possess all the properties listed below: (i) large number of heterogeneous NLG metrics, (ii) concurrent computation of more metrics at once, (iii) support for multiple references and/or predictions, (iv) meta-evaluation, and (v) visualization. Table 1 summarizes the discrepancies between NLG-METRICVERSE and related work.

3 Design Principles

NLG-METRICVERSE has been designed with five main principles in mind, which, we argue, can help researchers and practitioners in a number of ways.

Comprehensiveness Given the impressive pace at which the field is growing, comprehensiveness is imperative, with the ultimate goal of providing a unique, smooth, and up-to-date access point to all the most relevant NLG evaluation metrics disseminated in different streams of literature. We also comprise organization and consistency across the library, with a coherent interaction between modules and sub-modules. This principle revolves around consolidating an all-in-one community-driven library, integrating ready-to-use n-gram- and embedding-based metrics—supervised and unsupervised, trained and untrained, reference- and statistics-based, task-specific and general-purpose, sentence- and document-level. From this synergy, we hope to spur the adoption of newly proposed contributions, unleashing their potential and concretizing the view of Sellam et al. (2020), according to which "Machine Learning (ML) engineers should enrich their evaluation toolkits with more flexible, semantic-level metrics".

Ease-of-use The focus on simplicity is another key factor in fostering impact and usability, allowing users to write less code, reduce errors, and prototype faster. It is also meant to minimize the implementational burden and quickly move from papers to practical applications. We concentrate our efforts on designing an intuitive Application Programming Interface (API) accompanied by rich documentation with a curated list of executable notebooks and examples. This makes the software useful for both academia and industry.

Reproducibility Reproducibility is a core concept of utmost concern in ML and NLP, a prerequisite to trustworthiness. NLG evaluation exacerbates the problem even more, with well-known plagues like heavy undocumented preprocessing pipelines, non-transparent dataset selections, and concealed parameter settings (Post, 2018; Gao et al., 2021; Chen et al., 2022). A critical design objective of NLG-METRICVERSE is permitting experimental evaluation results to be seamlessly reproduced, promoting a fully detailed specification. In this way, users can simply integrate their original research into the shared codebase and fairly compare their solution with the existing literature. Besides serving for sound and consistent scientific research, reproducibility is a means to speed up the development of new metrics. When it comes to model-based metrics, transparency also applies to hardware setup, runtime measures, and CO2 impact.

Modularity In NLG-METRICVERSE, simplicity is sometimes bent in favor of modularity and reusability. This principle is essential for ensuring scalability and collaboratively bringing the codebase to maturity. An emphasis on module independence is maintained to guarantee the stand-alone usability of individual module functionalities and facilitate the learning of each library component.

Education One more principle is taking charge of an educational role. NLG-METRICVERSE is ideally suited to non-expert users, helping to sharpen their understanding. We believe that it is indispensable to democratize the field and gain greater
This paper was retracted. For more information, see https://aclanthology.org/2022.coling-1.306.
on learned representations and placed outside the backpropagation process. They typically refer to solutions based on regression, ranking, and classification tasks (e.g., COMET (Rei et al., 2020), FactCC (Kryscinski et al., 2020), BLEURT (Sellar et al., 2020), NUBIA (Kane et al., 2020)). Unsupervised metrics use a fixed set of heuristics and input features, such as n-gram overlapping, edit distance, static or contextualized embeddings. In this context, grammar-based measures do not rely on ground-truth references and try to quantify aspects like readability (i.e., the ease with which a reader can understand a passage) and grammaticality. To provide a concrete example, BERTScore is a context-free, reference-based, trained and unsupervised metric.

5 Main Modules

NLG-METRICVERSE is organized into three main modules: Metrics (§5.1), Meta-Evaluation (§5.2), and Visualization (§5.3). The library is intended to be a continuous and collaborative project, extended as new solutions become available. In what follows, we describe the features provided at the current stage of development. Figure 1 shows the operational representation of the modules and their interplay within the framework detailed in §4. NLG-METRICVERSE is in turn built on top of open-source libraries, including Datasets (Lhoest et al., 2021), NumPy (van der Walt et al., 2011), SciPy (Virtanen et al., 2020), and Matplotlib (Hunter, 2007). Where possible, metrics are implemented using canonical repositories released by authors.

5.1 Metrics

To construct a full-scale NLG evaluation library, the selection methodology is crucial to collect metrics with desired properties. We concentrate on four factors. (i) Diverse classes, supervision constraints, and evaluation tasks, as defined in §4. NLG is a versatile field; the input/output scenarios and evaluation strategies can vary from case to case. Sometimes, the predicted text is short and accompanied by human target references; other times, diversity is preferred; still different times, the generation is open-ended, long, and without references. (ii) Diverse application tasks. Metrics can apply to multiple NLG evaluation tasks or manage task-specific quality needs. Hence, we include a broad spectrum of real-world tasks to boost the relevance of our library. (iii) Eval dimension. Evaluation can be done by assessing different quality perspectives. Most existing metrics cover a small subset of these axes. Still, some of them—particularly the trainable ones—can handle several dimensions by requiring to maximize correlation with each type of judgment separately (Rei et al., 2020) or not (Yuan et al., 2021). (iv) Popularity. We give priority to the metrics prominently used in NLG research. Currently, 34 metrics are supported (see §A.1 for details); more solutions are under development. We tried to cover a balanced mixture of metrics and paid importance not to overweight a specific class. Future contributions can easily be integrated into NLG-METRICVERSE. We ensure the integrity of each metric within the codebase through automated tests.

Input Format. We design a unified metric input type, also handling a priority for candidate and reference texts (Table 3)—a feature as vital as neglected by current systems. In fact, there may exist multiple equally good outputs for the given input, and comparing against one gold reference can be erroneous. An extensive set of out-of-the-box data loaders takes the responsibility of processing the raw data from files and directories.

<table>
<thead>
<tr>
<th>Cardinality</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:1</td>
<td>preds = [p₁, ..., pₖ], refs = [r₁, ..., rₖ]</td>
</tr>
<tr>
<td>1:N</td>
<td>preds = [p₁]</td>
</tr>
<tr>
<td></td>
<td>refs = [[r₁₁, ..., r₁ₘ₁], ..., [rₖ₁, ..., rₖₘₖ]]</td>
</tr>
<tr>
<td>N:M</td>
<td>preds = [p₁₁, ..., pₖₙ₁], refs = [r₁₁, ..., rₖₙₙ]</td>
</tr>
<tr>
<td>Prefs only</td>
<td>preds = [p₁, ..., pₖ]</td>
</tr>
</tbody>
</table>

Table 3: Prediction-reference input formats.

Metrics Application. Evaluating artificial text requires just two lines of code: (i) create a Scorer object with the desired metrics; (ii) apply the Scorer object to the input data. So, many metrics may be executed in one go. During step (ii), the proper strategy for computing each metric is automatically selected depending on the recognized input format. If a prediction needs to be compared against multiple references, the user is left with the possibility to specify the aggregation strategy of preference through the reduce_fn parameter. For example, reduce_fn="max" considers only the prediction-reference pair with the highest score for each dataset instance. Inherently, NLG-METRICVERSE allows all NumPy function names and custom aggregation functions as well. An asynchronous execution with a separate process for each
metric can be specified to push efficiency and scalability (run_concurrent), bringing parallelism to the evaluation loop. Additionally, to contain the library size, we do not directly include all the packages required for running every supported metric, but we invite the user to install them if necessary. Figure 3 provides a practical example.

```python
1 scorer =
   NLGMetricverse(metrics=["bertscore", "bartscore"], run_concurrent=True)
2 score = scorer(preds, refs) # reduce_fn
```

Figure 3: Definition and application of a Scorer object for the concurrent evaluation of multiple metrics.

By employing the load_metric() function for step (i), NLG-METRICVERSE falls back to the Datasets implementation in case of metrics not yet supported. Consequently, our library englobes at least any metrics that the Datasets package has. When defining the Scorer, a maximum degree of freedom is retained to allow the setting of metric-specific hyperparameters and different instantiations of the same metric (Figure 4). Further, since metrics generally involve several hyperparameters and results can deviate significantly for other choices, we accompany the output with a config report (hyperparams setting, hardware setup, etc.) for increasing comparability and replicability.

The Scorer application is meant to return a dictionary containing each metric’s score(s), together with tracked performance metadata, including the computation time and CO2 emissions (measured with codecarbon (Schmid et al., 2021)).

```python
1 metrics = [
2 load_metric("bleu",
   resulting_name="bleu_1",
   compute_kwars={"max_order": 1}),
3 load_metric("bleu",
   resulting_name="bleu_2",
   compute_kwars={"max_order": 2}),
4 load_metric("rouge")
5 scorer = NLGMetricverse(metrics=metrics)
```

Figure 4: Definition and application of a Scorer object through the load_metric() function, encompassing two versions of BLEU with distinct hyperparameters.

### Metric Documentation and Search

NLP practitioners typically use automated metrics with a specific goal in mind, whether they are looking to answer a research question or develop a practical application system. To that end, they need to quickly identify which metric is most appropriate for the task at hand and understand how various attributes/properties might help with or, conversely, run contrary to their purpose. To let the user sift our NLG evaluation toolbox, we attach to each metric a set of structured tags (based on §4). Figure 5 exhibits APIs that allow users to list supported metrics and dig for those having preferred properties. We provide metric cards—inspired from aimed at evolving the Datasets ones—holding standardized information about metric functioning, technical aspects, output bounds, etc. Since a metric’s life continues beyond its initial release—from discovered weaknesses to newly found task adaptabilities, the metric card is conceived as a living document. The tags and metric cards are filled manually by the contributors who introduce the metrics to the library. The NLG-METRICVERSE community-driven nature and the GitHub-backend versioning provide an opportunity to keep the documentation up-to-date as further information comes to light.

```python
1 NLGMetricverse.list_metrics()
2 # All
3 NLGMetricverse.filter_metrics(
   category=Categories.Embedding,
   appl_task=AppTasks.DataToText)
4 # ["moverscore", "bleurt", "bartscore"]
5 NLGMetricverse.filter_metrics(
   trained=True, unsupervised=True,
   quality_dim=QualityDims.Factuality)
6 # ["bartscore"]
```

Figure 5: Taxonomy-guided metrics exploration.

### Custom Metric

NLG-METRICVERSE offers a flexible and uniform API for easily creating custom user-defined metrics. It only requires inheriting the MetricForNLG class (i.e., the common base class for each metric) and implementing the abstract functions linked to the possible input formats (Figure 6). We pursue the idea of enabling the user to create complex setups without superimposing constraints that may not suit future research.

```python
1 class CustomMetric (MetricForNLG):
2    def _compute_single_pred_single_ref(self, preds, refs, reduce_fn=None, **kwargs)
3        ...:
4    def _compute_single_pred_multi_ref ...
5    def _compute_multi_pred_multi_ref ...
```

Figure 6: Custom metric implementation.
5.2 Meta-Evaluation

With the ever-growing number of proposed metrics, evaluating NLG evaluation has notoriously become a compelling exigency. The \texttt{meta\_eval} module of NLG-METRICVERSE encompasses the most widely used methodologies for judging and comparing the effectiveness, reliability, and efficiency of automatic metrics. Few lines of code are sufficient to equitably assess a large number of published or prototype metrics on shared benchmarks.

Correlation Measures and Significance Tests

Examining a set of NLG metrics usually presupposes the computation of different correlation measures on paired data \[{(x_1, y_1), \ldots, (x_n, y_n)}\] depending on the goal and the relationship type between the two variables of interest \(X\) and \(Y\). We support four standard correlation coefficients:

- **Pearson Correlation** (Freedman et al., 2007), measures the \(X\)-\(Y\) linear dependence;
- **Spearman Correlation** (Zar, 2005), measures the \(X\)-\(Y\) monotonic relationships (whether linear or not);
- **Kendall’s \(\tau\)** (Kendall, 1938) measures the \(X\)-\(Y\) ordinal association (ranking preservation);
- **DARR** (Ma et al., 2018), a robust variant of Kendall’s \(\tau\) to account for potential noise in \(Y\) through pairs filtering.

We refer the reader to Sai et al. (2022) for an in-depth discussion on their differences and selection criteria. In all cases, coefficients take values in \([-1, 1]\), from low to high agreement, with 0 denoting total independence. To compute the statistical significance of the quantified dependency strength, NLG-METRICVERSE considers the p-value of a hypothesis test examining the evidence against the null hypothesis that “population correlation coefficient equals 0”. A smaller p-value means stronger evidence in favor of the alternative hypothesis, i.e., the population correlation is non-zero. The library also allows bootstrapping methods (Koehn, 2004) for rigorous pair-wise significance tests. Following previous works (Kilickaya et al., 2017; Novikova et al., 2017), we also incorporate the Williams’ test (Williams, 1959) for evaluating the significance between two dependent correlations sharing one variable (i.e., \(X_1\), \(X_2\), and \(Y\)).

Metric-Human Correlation

One of the primary goals of \texttt{meta\_eval} is to analyze the extent to which different automatic evaluation metrics agree with human judgments (Figure 7). To do so, we provide tools for constructively computing metric-human correlations on popular benchmarks or custom user ground truths, where \(X\) and \(Y\) correspond to metric and human scores, respectively. As for benchmarking, we underline the urgency of standardized datasets containing <context, prediction, reference, human scores> tuples for multiple tasks, quality dimensions, and languages. The development of NLG evaluation metrics relies on their availability, both for training and evaluation purposes. Unfortunately, despite the evolving interest, there is still a scarcity of contributions in this direction. Currently, we use the annual public records from the WMT Metrics Shared Task (Bojar et al., 2017)—the largest collection of human ratings at the time of writing (i.e., human-annotated machine translation pairs).

\begin{verbatim}
metric_human_correlation(preds, refs, 
    metrics=load_metric("rouge"), 
    compute_kwargs={"rouge_types": ["rougel"]}, 
    human_scores=Benchmarks.WMT17, 
    corrs=[CorrelationMeasures.Pearson])
\end{verbatim}

Figure 7: Segment-level metric-human correlation scatterplot. ROUGE vs. human scores on WMT17.

Metric-Metric Comparison

On the trail of the most frequent evaluation setups used in literature, we supply functional features for checking out the behavior of many models side-by-side. In fact, metrics are best understood when compared to each other on common datasets. This comparison refers to performance aspects (e.g., computation time, \(CO2\) impact for model-based metrics) and correlations (i.e., input-output similarities). Ultimately, NLG-METRICVERSE showcases the results with a set of meaningful charts intended to embolden scientific documentation (examples in Figure 9).
5.3 Visualization

In contrast to human evaluation, automatic metrics generally assign a single score to a given hypothesis, and it is often not clear which quality perspective this score captures or corresponds to; ergo, they are difficult to interpret (Sai et al., 2022). Score uninterpretability not only applies to contemporary model-based solutions but also to historical n-gram approaches (Zhang et al., 2004). More generally, visualization tools have become a cornerstone of explainability research in NLP. To increase the transparency of NLG evaluation metrics, we provide static and interactive visual tools for understanding why certain scores are produced. Visually inspecting internal mechanisms is particularly useful in instances when metrics disagree on. The interactive visualizations are built using web technologies manipulated through D3.js (Bostock et al., 2012). Supported ones include soft and hard alignments from MOVERScore and BERTScore (Figure 8).

![Figure 8: Examples of plots for visual metric analysis.](https://aclanthology.org/2022.coling-1.306)

(a) MOVERScore, IDF-weighted n-gram soft-alignment.

(b) BERTScore, Color-coded cosine similarity word matching.

6 Case Study: Graph-Augmented Biomedical Abstractive Summarization

In this section, we use NLG-METRICVERSE to examine the summaries generated by a language model infused with semantic parsing graphs. Injecting explicit semantic structures—like events (Frisoni et al., 2021, 2022), abstract meaning representations (AMRs) (Banarescu et al., 2013), and corpus-level knowledge (Frisoni et al., 2020; Frisoni and Moro, 2020)—is a new trend followed by the NLP community to overcome lexical superficiality and draw a complementary path to architectural scaling, fundamental in low-resource settings (Moro and Ragazzi, 2022). Graph-augmented methods unlock a higher level of abstraction and more accurate emulation of human interpretation, rewriting, and paraphrasing. Faced with semantic-driven models, researchers must avoid being confined to traditional overlap-based metrics and monolithic quality dimensions, thus outlining a valuable testbed for our library.

6.1 Experimental Setup

We employ COGITO-ERGO-SUMM (Frisoni et al., 2022), a language model for biomedical single-document summarization, enhanced by AMRs and structured representations of factual evidence extracted from the source text. By employing the same hyperparameters proposed by the authors, we train and evaluate the neural network on CDSR (Guo et al., 2021)—a dataset designed for health literacy, where the training, validation, and test sets contain 5178, 500, and 999 samples, respectively. To quantitatively inspect model performance on the test set, we apply NLG-METRICVERSE for computing ROUGE-1/2/L (F1), BERTScore, BARTScore (Recall), Abstractness, and Repetitiveness. Additionally, since CDSR targets the accessibility of the biomedical literature, we calculate readability scores: Gunning Fog Index, Flesch-Kincaid Reading Ease, Coleman-Liau Index. See A.1 for details about metrics functioning, and A.2 for replicability. To better gauge summary quality and compare metrics’ effectiveness, we conduct a human evaluation study. We randomly select 30 test set instances, and invite 3 expert annotators to score generated summaries in conformity with four independent perspectives, each measured on a Likert scale from 1 (worst) to 5 (best): (i) informativeness, i.e., conveying salient content; (ii) factualness, i.e., being faithful with respect to the article; (iii) fluency, i.e., being fluent, grammatical, and coherent; (iv) succinctness, i.e., non containing redundant and unnecessary information.

6.2 Results

Figure 9 reports human and automatic evaluation results, together with computation times, metric-metric, and metric-human correlations (Pearson).
Figure 9: Abstractive summarization analysis through NLG-METRICVERSE.

Human scores are averaged for each dimension; the mean Kendall coefficient among all evaluators’ inter-rater agreement is 0.16. We observe that the abstractive and semantically-consistent nature of the model is not appreciable by the ROUGE scores alone. The highest correlations with human judgment are achieved by BERTScore, Abstractness, and Flesch-Kincaid—especially according to factualness and succinctness (see A.2). These results prove that the model tends to be more factual when it re-frames the target concept units, further testifying the inadequacy of overlap-based metrics. Notably, in contrast to other model-based metrics like BERTScore, BARTScore appears significantly slower (72× compared to ROUGE).

7 Conclusion

The NLG evaluation community demands efforts toward making research more transparent, reproducible, and open. Easy access to a wide variety of automatic metrics and related features holds a lot of potential. A central hub would democratize research, increase comparability, mitigate the computational/implementational burden, and hopefully steer innovation to more robust contributions. In fact, researchers would be able to evaluate their NLG systems at scale without being limited to very few metrics whose code is easily available. They would also be able to critically examine existing metrics, perform white-box attacks, or carefully craft adversarial examples.

With NLG-METRICVERSE, we take an important step towards a single, unified, coherent, end-to-end, and easily extendable framework for NLG evaluation. A solid reference point and shared resource for researchers and practitioners working in the area. Being a community-driven effort, we plan in both the near and medium terms to support more recent task-specific metrics, benchmarks, meta-evaluation techniques for robustness, and skew factor analyses. We also intend to include more document-level measures. We hope that this library may trigger a positive reinforcement loop within our community, nudging it to explore the metric universe.

References


Satanjeev Banerjee and Alon Lavie. 2005. METEOR:
This paper was retracted. For more information, see https://aclanthology.org/2022.coling-1.306.


This paper was retracted. For more information, see https://aclanthology.org/2022.coling-1.306.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Technique</th>
<th>Property</th>
<th>Appl. Tasks</th>
<th>Trained</th>
<th>Unsupervised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gunning Fog Index</td>
<td>G</td>
<td>readability test for English writing: count of sentences, words, and complex words consisting of three or more syllables in the text</td>
<td>SUM</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Flesch-Kincaid</td>
<td>G</td>
<td>the most widely used readability test for English writing; two versions (Flesch Reading-Ease and Flesch-Kincaid Grade Level)</td>
<td>SUM</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Coleman-Liau Index</td>
<td>G</td>
<td>character-based readability test for English writing</td>
<td>SUM</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Accuracy</td>
<td>N</td>
<td>proportion of correct predictions among the total number of cases processed</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Pedregosa et al. 2011</td>
<td>N</td>
<td>fraction of correctly labeled positive examples out of all of the examples that were labeled as positive</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Recall</td>
<td>N</td>
<td>fraction of positive examples correctly labeled by the model as positive</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>F1</td>
<td>N</td>
<td>harmonic mean of the precision and recall</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>MER</td>
<td>N</td>
<td>% words incorrectly predicted and inserted (match error rate)</td>
<td>SUM</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Abstractionness</td>
<td>N</td>
<td>% novel n-grams in the predictions, compared to the references</td>
<td>SUM</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Repetitiveness</td>
<td>N</td>
<td>average number of n-grams with at least one repetition in the generated sequences</td>
<td>SUM</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Coverage</td>
<td>N</td>
<td>% summary words present in the source text</td>
<td>SUM</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Density</td>
<td>N</td>
<td>average length of extracted fragments which every word from the summary belongs to</td>
<td>SUM</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Compression</td>
<td>N</td>
<td>ratio between the length of the original text and the length of the generated abstract</td>
<td>SUM</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>BLEU</td>
<td>N</td>
<td>n-gram precision</td>
<td>MT, IC, DG, QG, RG</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Papineni et al., 2002</td>
<td>N</td>
<td>n-gram precision w/ IDF-weighted n-grams</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Doddington 2002</td>
<td>N</td>
<td>n-gram precision w/ smoothing</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>ORANGE (SentBLEU)</td>
<td>N</td>
<td>n-gram precision</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Lin and Och 2004</td>
<td>N</td>
<td>n-gram precision</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>ROUGE</td>
<td>Lin, 2004</td>
<td>n-gram recall</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>WER</td>
<td>N</td>
<td>% of insert, delete, replace</td>
<td>MT, SR</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Morris et al. 2004b</td>
<td>N</td>
<td>n-gram harmonizing metric with paraphrase knowledge (e.g., stemming, synonym, and penalty factor for fragmented matches)</td>
<td>MT, IC, DG</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Banerjee and Lavie 2005</td>
<td>N</td>
<td>cosine similarity between TF-IDF weighted n-grams</td>
<td>IC</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>CIDEr</td>
<td>Vedantam et al. 2015</td>
<td>n-gram based metric w/ translation edit distance (e.g., WER + shift movement as extra editing step)</td>
<td>IC</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>TERR</td>
<td>Snover et al. 2006</td>
<td>translation edit distance w/ shift movement as extra editing step</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>ChRf(++)</td>
<td>Popović 2017</td>
<td>character-level precision and recall</td>
<td>MT, IC, SUM</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>WMD</td>
<td>Kuwata et al. 2015</td>
<td>earth mover’s distance on words</td>
<td>IC, SUM</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>SMS</td>
<td>Clark et al. 2019</td>
<td>earth mover’s distance on sentences</td>
<td>IC, SR, SUM</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>CharacterTER</td>
<td>Wang et al. 2016</td>
<td>character-level TER</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>SacreBLEU</td>
<td>N</td>
<td>standardized BLEU</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Post 2018</td>
<td>N</td>
<td>METEOR w/ copy knowledge and syntactic-level paraphrase matching</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>METEOR++</td>
<td>Guo and Hu 2019</td>
<td>METEOR w/ copy knowledge and syntactic-level paraphrase matching</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 5: NLG-METRICVERSE supported metrics for the v1.0.0 release, in ascending order of publication. We use the following abbreviations for different techniques and features: G – Grammar-based, N – N-gram-based, D – Distance-based, E – Embedding-based, S – Statistics-based. For tasks, SUM – Summarization, MT – Machine Translation, SR – Speech Recognition, IC – Image Captioning, DG – Document or Story Generation, QG – Query Generation, RG – Dialogue Response Generation, D2T – Data-to-Text, TC – Text Completion; we only list the ones justified by the original paper or by the first NLG application.
This paper was retracted. For more information, see https://aclanthology.org/2022.coling-1.306.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Technique</th>
<th>Property</th>
<th>Appl. Tasks</th>
<th>Trained</th>
<th>Unsupervised</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOVERScore</td>
<td>E</td>
<td>IDF-weighted n-gram soft-alignment (WMD generalization) via contextualized embeddings; it computes the minimum cost of transforming the generated text to the reference text, taking into account Euclidean distance between vector representations of n-grams, as well as their document frequencies</td>
<td>MT, SUM, D2T, IC</td>
<td>✓</td>
<td>✓ ELMo/BERT</td>
</tr>
<tr>
<td>EED</td>
<td>D</td>
<td>Levenshtein distance + jump operation</td>
<td>MT</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>COMET</td>
<td>E</td>
<td>multilingual-MT human judgment predictions through pre-trained cross-lingual encoders (word embeddings) + pooling layers (sentence embeddings) + feed-forward regressor or triplet margin loss depending on the judgment type (real-value or relative ranking)</td>
<td>MT</td>
<td>✓</td>
<td>× XML-RoBERTa</td>
</tr>
<tr>
<td>FactCC(X)</td>
<td>E</td>
<td>weakly-supervised document-sentence factual consistency evaluation based on BERT’s [CLS] embedding</td>
<td>SUM</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>BLEURT</td>
<td>E</td>
<td>robust human score prediction based on fine-tuning a BERT model with an additional pre-training scheme characterized by millions of synthetic reference-candidate pairs and lexical-/semantic-level tasks combined through an aggregated loss</td>
<td>MT, D2T</td>
<td>✓</td>
<td>× BERT-end-to-end</td>
</tr>
<tr>
<td>NUBIA</td>
<td>E</td>
<td>human score prediction with three modules: neural feature extractor on reference-hypothesis pairs (multiple pre-trained transformers capturing semantic similarity, logic entailment, sentence intelligibility) + aggregator (features→quality score mapping) + calibrator</td>
<td>MT, IC</td>
<td>✓</td>
<td>× GPT2-end-to-end</td>
</tr>
<tr>
<td>BERTScore</td>
<td>E</td>
<td>IDF-weighted n-gram hard-alignment via contextualized embeddings</td>
<td>MT, IC</td>
<td>✓</td>
<td>✓ BERT</td>
</tr>
<tr>
<td>BARTScore</td>
<td>E</td>
<td>multi-perspective evaluation as text generation via a pre-trained seq2seq model to measure how likely hypothesis and reference are paraphrased according to the probability of one giving the other</td>
<td>MT, SUM, D2T, IC</td>
<td>✓</td>
<td>✓ BART</td>
</tr>
<tr>
<td>Perplexity</td>
<td>E</td>
<td>how likely a model is to generate the input text sequence</td>
<td>SR</td>
<td>✓</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>PRISM</td>
<td>E</td>
<td>sequence-to-sequence paraphraser to score MT system outputs conditioned on their respective human references</td>
<td>TC</td>
<td>✓</td>
<td>✓ GPT2 Grover</td>
</tr>
<tr>
<td>MAUVE</td>
<td>E, D</td>
<td>comparison measure for open-ended text generation w/ divergences in a quantized embedding space</td>
<td>TC</td>
<td>✓</td>
<td>✓ GPT2 Grover</td>
</tr>
</tbody>
</table>

Table 6: Table 5 continuation.

Figure 10: Pearson correlations between automatic metrics and human annotations for each quality dimension inspected in the case study, i.e., informativeness, factualness, fluency, succinctness.