RoMe: A Robust Metric for Evaluating Natural Language Generation

Anonymous ACL submission

Abstract

Evaluating Natural Language Generation (NLG) systems is a challenging task. Firstly, the metric should ensure that the generated hypothesis reflects the reference’s semantics. Secondly, it should consider the grammatical quality of the generated sentence. Thirdly, it should be robust enough to handle various surface forms of the generated sentence. Thus, an effective evaluation metric has to be multifaceted. In this paper, we propose an automatic evaluation metric incorporating several core aspects of natural language understanding (language competence, syntactic and semantic variation). Our proposed metric, RoMe, is trained on language features such as semantic similarity combined with tree edit distance and grammatical acceptability, using a self-supervised neural network to assess the overall quality of the generated sentence. Moreover, we perform an extensive robustness analysis of the state-of-the-art methods and RoMe. Empirical results suggest that RoMe has a stronger correlation to human judgment over state-of-the-art metrics in evaluating system-generated sentences across several NLG tasks.

1 Introduction

Automatic generation of fluent and coherent natural language is a key step for human-computer interaction. Evaluating generative systems such as text summarization, dialogue systems, and machine translation is challenging since the assessment involves several criteria such as content determination, lexicalization, and surface realization (Liu et al., 2016; Dale and Mellish, 1998). For assessing system-generated outputs, human judgment is considered to be the best approach. Obtaining human evaluation ratings, on the other hand, is both expensive and time-consuming. As a result, developing automated metrics for assessing the quality of machine-generated text has become an active area of research in NLP.

The quality estimation task primarily entails determining the similarity between the reference and hypothesis as well as assessing the hypothesis for grammatical correctness and naturalness. Widely used evaluation metrics such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and ROUGE (Lin, 2004) which compute the word-overlaps, were primarily designed for evaluating machine translation and text summarization systems. Word-overlap based metrics, on the other hand, are incapable of capturing the hypotheses’ naturalness and fluency. Furthermore, they do not consider the syntactic difference between reference and hypothesis. In a different line of research, word mover distance (WMD) (Kusner et al., 2015a), BERTScore (Zhang et al., 2020) and MoverScore (Zhao et al., 2019) compute word embedding based similarity for evaluating system-generated texts. Although these metrics employ the contextualized representation of words, they do not take the grammatical acceptability of the hypothesis and the syntactical similarity to the reference into account.

To address these shortcomings, we propose RoMe, an automatic and robust metric for evaluating NLG systems. RoMe employs a neural classifier that uses the generated sentence’s grammatical, syntactic, and semantic qualities as features to estimate the quality of the sentence. Firstly, it calculates the earth mover’s distance (EMD) (Rubner et al., 1998) to determine how much the hypothesis differs from the reference. During the computation of EMD, we incorporate hard word alignment and soft-penalization constants to handle various surface forms of words in a sentence, such as repeated words and the passive form of a sentence. Secondly, using a semantically enhanced tree edit distance, the difference in syntactic structures between the reference and hypothesis sentences is quantified. Thirdly, the metric incorporates a binary classifier to evaluate the grammatical accept-
The Earth Mover’s Distance (EMD) estimates the amount of work required to transform a probability distribution into another (Rubner et al., 1998). Inspired by the EMD, in NLP the transportation problem is adopted to measure the amount of work required to match the system generated hypothesis sentence with the reference sentence (Kusner et al., 2015b; Zhao et al., 2019). Let us define the reference as $\mathcal{R} = \{r_1, r_2, \ldots, r_p\}$ and the hypothesis as $\mathcal{H} = \{h_1, h_2, \ldots, h_q\}$, where $r_i$ and $h_j$ indicates the $i$-th and $j$-th word of the reference and hypothesis, respectively. The weight of the word $r_i$ and $h_j$ are denoted as $m_i$ and $n_j$ respectively. Then, the total weight distribution of $\mathcal{R}$ and $\mathcal{H}$ is $m_{\sum} = \sum_{i=1}^{p} m_i$ and $n_{\sum} = \sum_{j=1}^{q} n_j$ respectively. Here, the sentence-level and normalized TF-IDF score of a word is considered as the word’s weight. Formally, EMD can be defined as:

$$EMD(\mathcal{H}, \mathcal{R}) = \frac{\min_{f_{ij} \in F(\mathcal{H}, \mathcal{R})} \sum_{i=1}^{p} \sum_{j=1}^{q} d_{ij} f_{ij}}{\min(m_{\sum}, n_{\sum})}$$

where $d_{ij}$ is the distance between the words $r_i$ and $h_j$ in the space and $F(\mathcal{H}, \mathcal{R})$ is a set of possible flows between the two distribution that the system tries to optimize. In Equation 1, $EMD(\mathcal{H}, \mathcal{R})$ denotes the amount of work required to match the hypothesis with the reference. The optimization is done following four constraints:

1. $f_{ij} \geq 0$ for $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, q$,
2. $\sum_{j=1}^{q} f_{ij} = m_i$ for $i = 1, 2, \ldots, p$,
3. $\sum_{i=1}^{p} f_{ij} = n_j$ for $j = 1, 2, \ldots, q$,
4. $\sum_{i=1}^{p} \sum_{j=1}^{q} f_{ij} = \min(m_{\sum}, n_{\sum})$

Figure 1 depicts the EMD for a given hypothesis-reference pair.

### 2 Preliminaries

#### 2.1 Earth Mover’s Distance

The Earth Mover’s Distance (EMD) estimates the amount of work required to transform a probability distribution into another (Rubner et al., 1998). Inspired by the EMD, in NLP the transportation problem is adopted to measure the amount of work required to match the system generated hypothesis sentence with the reference sentence (Kusner et al., 2015b; Zhao et al., 2019). Let us define the reference as $\mathcal{R} = \{r_1, r_2, \ldots, r_p\}$ and the hypothesis as $\mathcal{H} = \{h_1, h_2, \ldots, h_q\}$, where $r_i$ and $h_j$ indicates the $i$-th and $j$-th word of the reference and hypothesis, respectively. The weight of the word $r_i$ and $h_j$ are denoted as $m_i$ and $n_j$ respectively. Then, the total weight distribution of $\mathcal{R}$ and $\mathcal{H}$ is $m_{\sum} = \sum_{i=1}^{p} m_i$ and $n_{\sum} = \sum_{j=1}^{q} n_j$ respectively. Here, the sentence-level and normalized TF-IDF score of a word is considered as the word’s weight. Formally, EMD can be defined as:

$$EMD(\mathcal{H}, \mathcal{R}) = \frac{\min_{f_{ij} \in F(\mathcal{H}, \mathcal{R})} \sum_{i=1}^{p} \sum_{j=1}^{q} d_{ij} f_{ij}}{\min(m_{\sum}, n_{\sum})}$$

where $d_{ij}$ is the distance between the words $r_i$ and $h_j$ in the space and $F(\mathcal{H}, \mathcal{R})$ is a set of possible flows between the two distribution that the system tries to optimize. In Equation 1, $EMD(\mathcal{H}, \mathcal{R})$ denotes the amount of work required to match the hypothesis with the reference. The optimization is done following four constraints:

1. $f_{ij} \geq 0$ for $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, q$,
2. $\sum_{j=1}^{q} f_{ij} = m_i$ for $i = 1, 2, \ldots, p$,
3. $\sum_{i=1}^{p} f_{ij} = n_j$ for $j = 1, 2, \ldots, q$,
4. $\sum_{i=1}^{p} \sum_{j=1}^{q} f_{ij} = \min(m_{\sum}, n_{\sum})$

Figure 1 depicts the EMD for a given hypothesis-reference pair.

#### 2.2 Syntactic Similarity and Tree Edit Distance

In computational linguistics, dependency and constituency trees are used to represent syntactic dependencies between words in a sentence. Unlike the constituency tree, a dependency tree can represent non-adjacent and non-projective dependencies in a sentence, which frequently appear in spoken language and noisy text. That leads us to prefer dependency trees over constituency trees for evaluating NLG output.
Formally, a dependency tree is a set of nodes
Ω = \{w_0, w_1, ..., w_k\} and a set of dependency
links \( G = \{g_0, g_1, ..., g_k\} \), where \( w_0 \) is the imagi-
nary root node and \( g_i \) is an index into \( \Omega \) represent-
ing the governor of \( w_i \). Every node has exactly
one governor except for \( w_0 \), which has no gover-
nor (Hall and Novák, 2010). Syntactic similarity
between a pair of dependency trees can be esti-
mated using several methods, such as graph cen-
tralities and Euclidean distances (Oya, 2020). In
our work, we exploit the Tree Edit Distance (TED)
algorithm (Zhang and Shasha, 1989) to estimate
syntactic similarity between reference and hypothe-
sis. TED is typically computed on ordered labeled
trees and can thus be used to compare dependency
trees. The edit operations performed during the
comparison of parsed dependency trees include
Change, Delete, and Insert.

![Figure 2: Visualization of the required edit operations to transform \( T_H \) to \( T_R \). The operations corresponds to the following sequence: delete(node with label \( c \)), insert(node with label \( c \)).

Let's consider \( T_H \) and \( T_R \) be the parsed depen-
dency trees of the hypothesis and reference, respec-
tively. The operations required to transform one
tree into another are visualized in Figure 2. In TED,
an exact match between the nodes of the compared
trees is performed to decide if any edit operation
is required. In this work, the syntactic difference
between hypothesis and reference is determined by
the output of TED, which specifies the total number
of edit operations.

3 RoMe

In RoMe, a neural network determines the final
evaluation score given a reference-hypothesis pair.
The network is trained to predict the evaluation
score based on three features: semantic similar-
ity computed by EMD, enhanced TED, and the
grammatical acceptability score. We explain these
features in the following subsections.

3.1 Earth Mover’s Distance based Semantic
Similarity

During the computation of EMD, we employ Se-
matic Word Alignment and Soft-penalization tech-
niques to tackle repetitive words and passive forms
of a sentence. We compute a distance matrix and a
flow matrix as described below and finally obtain
EMD utilizing Equation 1.

**Semantic Word Alignment.** We first align the
word pairs between reference and hypothesis based
on their semantic similarities. The alignment is
performed by computing all paired cosine similar-
ities while taking word position information into
account, as in (Echizen-ya et al., 2019). In contrast
to (Echizen-ya et al., 2019), we use contextualized
pre-trained word embedding from the language
model ALBERT (Lan et al., 2020). ALBERT uses
sentence-order prediction loss, focusing on mod-
eling inter-sentence coherence, which improves
multi-sentence encoding tasks.

The word alignment score is computed as fol-
lows:

\[
A(r_i, h_j) = \frac{r_i \cdot h_j}{\|r_i\| \|h_j\|} \cdot \frac{|q(i+1) - p(j+1)|}{pq}
\]

where \( r_i \) and \( h_j \) denote the contextualized word
embedding of \( r_i \) and \( h_j \), respectively. The first
part of the right side of the equation computes the
cosine similarity between \( r_i \) and \( h_j \), and the second
part calculates the relative position information as
proposed in (Echizen-ya et al., 2019).

![Figure 3: An example word alignment matrix for the reference sentence: "tesla motors is founded by elon musk" and its passive form: "elon musk founded tesla motors" is illustrated here.

Figure 3 depicts a matrix of word alignment
scores generated on an example pair of sentences.
This alignment strategy fails to handle repetitive
words where a word from the hypothesis may get
aligned to several words in the reference (see Fig-
ure 4). To tackle such cases, we restrict the word
alignment by imposing a hard constraint. In the
hard constraint, we prevent the words in the hypo-
thesis from getting aligned to multiple words in the
reference as illustrated by the dotted arrows in Fig-
ure 4. We denote the resulting set of hard-aligned
word pairs as \( A_{hc} \).
The semantic similarity between hypothesis and reference is denoted as $F_{sem} = 1.0 - EMD$. The normalized value of EMD is used to calculate $F_{sem}$.

### 3.2 Semantically Enhanced TED

To estimate the difference between the syntactic structures of reference and hypothesis, we extend the TED algorithm (Zhang and Shasha, 1989). The original TED algorithm performs edit operations based on an exact match between two nodes in the dependency trees of hypothesis and reference. In this work, we modify the TED algorithm and compute a word embedding-based cosine similarity to establish the equivalence of two nodes. Two nodes are considered equal if the cosine similarity of their embedding representations exceeds the threshold $\theta$. This allows the semantically enhanced TED to process synonyms and restricts it from unnecessary editing of similar nodes. We call the resulting algorithm TED-SE. The normalized value of TED-SE is denoted as $F_{ted}$. We compute TED-SE over the lemmatized reference and hypothesis since lemmatized text exhibits improved performance in such use cases (Kutuzov and Kuzmenko, 2019). The lemmatizer and dependency parser from Stanza (Qi et al., 2020) is utilized to obtain the tree representation of the text. More experimental detail such as the tree-transformation is provided in Appendix A.1.

### 3.3 Grammatical Acceptability Classification

Linguistic competence assumes that native speakers can judge the grammatical acceptability of a sentence. However, system-generated sentences are not always grammatically correct or acceptable. Therefore, we train a binary classifier on the Corpus of Linguistic Acceptability (CoLA) (Warstadt et al., 2018), predicting the probability that the hypothesis is grammatically acceptable. CoLA is a collection of sentences from the linguistics literature with binary expert acceptability labels containing over 10k examples (Warstadt et al., 2018). The classifier is based on BERT-large (Devlin et al., 2019) and trained to optimize binary cross-entropy loss. A text sequence is fed as input and as output the classifier produces the class membership probability (grammatically acceptable, grammatically unacceptable). The model achieves an accuracy of 80.6% on the out-of-domain CoLA test set (Warstadt et al., 2018, p. 8). We denote the score from the classifier as the feature $F_g$, which is used to train a neural network (see §3.4). Model parameters can be found in Appendix A.2.
3.4 Final Scorer Network

A feed-forward neural network takes the previously computed features as input and learns a function \( f(F_{\text{sem}}; F_{\text{text}}; F_g) \) in the final step, yielding a final output score in the \([0, 1]\) interval. The output score is regarded as the overall quality of the hypothesis. Following a self-supervised paradigm, the network is trained on artificially generated training samples from the KELM dataset (Agarwal et al., 2021). We randomly choose 2500 sentence pairs from the KELM dataset and generate 2500 more negative samples by randomly augmenting the sentences using TextAttack (Morris et al., 2020) and TextFooler (Jin et al., 2020). Following a similar approach, we additionally generate 1000 test sentence pairs from the KELM dataset. Overall, we then have 5000 training and 1000 test examples. The network is a simple, two-layered feed-forward network optimized with stochastic gradient descent using a learning rate of 1e-4.

4 Experiments and Analysis

4.1 Data

To assess RoMe’s overall performance, first, we benchmark on two language generation datasets, BAGEL (Mairesse et al., 2010) and SFHOTEL (Wen et al., 2015), containing 404 and 796 data points respectively. Each data point contains a meaning representation (MR) and a system generated output. Human evaluation scores of these datasets are obtained from (Novikova et al., 2017). Furthermore, we evaluate dialogue system’s outputs on Stanford in-car dialogues (Eric et al., 2017) containing 2,510 data points and the soccer dialogue dataset (Chaudhuri et al., 2019) with 2,990 data points. Each data point of these datasets includes a user query, a reference response, and a system response as a hypothesis. Each of the datasets comes with three system outputs for each of the dialogue data points. We use the human annotated data provided by (Chaudhuri et al., 2021). Moreover, we evaluate the metric on the system generated outputs from the NLG2017 challenge (Shimorina et al., 2018).

Finally, we randomly sample 200 data points from KELM (Agarwal et al., 2021) and perturb them with adversarial text transformation techniques, for conducting robustness analysis. Three annotators participated in a data annotation process (two of them are from a CS and one from a non-CS background), where they annotated the perturbed data. We provided the annotators with an annotation tool where the tool displays the reference sentence and the system output for each data point. The annotators were asked to choose a value from a range of \([1, 3]\), for each of the categories: \textit{Fluency, Semantic Correctness, and Grammatical correctness}. In this case, the values stand for 1: \textit{poor}, 2: \textit{average}, and 3: \textit{good}. The overall inter-annotator agreement score, \(\kappa\) is 0.78. The annotation tool and its interface are discussed in detail in Appendix A.3.

4.2 Baselines

We select both the word-overlap and embedding-based metrics as strong baselines. For the experiment and robustness analysis we choose BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), BERTScore (Zhang et al., 2020) and Mover-
Table 2: Component-wise qualitative analysis.

<table>
<thead>
<tr>
<th>Text</th>
<th>BLEU</th>
<th>BERTScore</th>
<th>MoverScore</th>
<th>RoMe</th>
</tr>
</thead>
<tbody>
<tr>
<td>R: Munich is located at the southern part of Germany. H: Munich is situated in the south of Germany.</td>
<td>0.83</td>
<td>1.0</td>
<td>0.94</td>
<td>0.80</td>
</tr>
<tr>
<td>R: Tesla motors is founded by Elon Musk. H: Elon Musk has founded Tesla Motors.</td>
<td>0.70</td>
<td>0.85</td>
<td>0.96</td>
<td>0.69</td>
</tr>
<tr>
<td>R: Elon musk has founded tesla motors. H: Elon elon elon elon founded tesla tesla.</td>
<td>0.01</td>
<td>0.50</td>
<td>0.17</td>
<td>0.11</td>
</tr>
</tbody>
</table>

Table 3: Qualitative analysis.

<table>
<thead>
<tr>
<th>Dialogue dataset</th>
<th>Model</th>
<th>correlation</th>
<th>SentBLEU</th>
<th>METEOR</th>
<th>BERTScore</th>
<th>MoverScore</th>
<th>RoMe</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-car dialogue</td>
<td>MemSeq</td>
<td>ρ</td>
<td>0.07</td>
<td>0.35</td>
<td>0.40</td>
<td>0.29</td>
<td>0.31</td>
</tr>
<tr>
<td>GLMP</td>
<td>ρ</td>
<td>0.04</td>
<td>0.29</td>
<td>0.32</td>
<td>0.31</td>
<td>0.32</td>
<td>0.11</td>
</tr>
<tr>
<td>DialoGPT</td>
<td>ρ</td>
<td>0.17</td>
<td>0.60</td>
<td>0.62</td>
<td>0.73</td>
<td>0.78</td>
<td>0.11</td>
</tr>
<tr>
<td>Soccer dialogue</td>
<td>MemSeq</td>
<td>ρ</td>
<td>0.03</td>
<td>0.08</td>
<td>0.08</td>
<td>0.11</td>
<td>0.11</td>
</tr>
<tr>
<td>GLMP</td>
<td>ρ</td>
<td>0.02</td>
<td>0.08</td>
<td>0.03</td>
<td>0.12</td>
<td>0.14</td>
<td>0.11</td>
</tr>
<tr>
<td>DialoGPT</td>
<td>ρ</td>
<td>0.04</td>
<td>0.26</td>
<td>0.31</td>
<td>0.39</td>
<td>0.43</td>
<td>0.11</td>
</tr>
</tbody>
</table>

Table 4: Metrics Spearman’s correlation coefficient (ρ) with human judgment on dialogue datasets.

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Correlation (ρ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoMe with EMDWord</td>
<td>64.8</td>
</tr>
<tr>
<td>+ EMDWord</td>
<td>66.0</td>
</tr>
<tr>
<td>+ EMDChar</td>
<td>66.9</td>
</tr>
<tr>
<td>+ TED-SE</td>
<td>67.1</td>
</tr>
<tr>
<td>+ Grammar</td>
<td>70.1</td>
</tr>
</tbody>
</table>

Table 5: Ablation Study.

4.3 Results

Table 1 shows the performance of different metrics on data to language generation dataset (BAGEL and SFHOTEL). In both the BAGEL and SFHOTEL, a meaning representation (MR), for instance inform(name='hotel drisco',price_range='pricey') is given as a reference sentence where the system output is: the hotel drisco is a pricey hotel, in this case. Although, RoMe outperformed the baseline metrics in evaluating the informativeness, naturalness and quality score, the correlation scores are still low with respect to the human judgement, because the MR is the reference statement in this case. For all the experiments, we take the normalized human judgement scores. We firstly evaluate our model by using Fasttext (Bojanowski et al., 2017) word embedding. We notice a significant improvement in results when we replace the Fasttext embedding with contextualized word embedding obtained from BERT (Devlin et al., 2019). Furthermore, we experiment with multiple language models and finally, we reach to our best performing model with ALBERT-large (Lan et al., 2020). In all the experiments, we report the results of RoMe, using ALBERT-large (Lan et al., 2020). In Table 1, WMD and SDM refer to word mover distance and sentence mover distance respectively, used in MoverScore. We report the results of WDM and SMD from (Zhao et al., 2019).

Table 4 shows the evaluation results on dialogue datasets. In case of in-car dataset, all the non-word-overlap metric achieved a better correlation score than the word-overlap based metrics. Because, in dialogue systems, the generated responses are evaluated based on the overall semantic meaning and the correctness of the response, the cases where the word-overlap-based metrics failed. Overall, RoMe achieves higher correlation scores in both the in-car and soccer dialogue datasets in evaluating multiple dialogue system outputs.

Finally, we evaluate outputs from 9 different systems participated in the NLG2017 challenge and report the correlation scores in Table 6. Although, RoMe achieves the best correlation in most of the cases, we also notice a comparable and in some cases better results achieved by the MoverScore (Zhao et al., 2019).

To further delve down into the metrics, a correlation graph is plotted in Figure 5. The graph is constructed from the scores of the metrics from the BAGEL dataset. As observed from the correlation graph, we can infer that our proposed metric, RoMe correlates highly with the MoverScore. However, since RoMe handles both the syntactic and semantic properties of the text it achieved better results in all the datasets across different NLG tasks.

4.4 Ablation Study

We conduct an ablation study to investigate the impact of the RoMe’s components on its overall performance. Table 5 shows the incremental improvement in Spearman’s correlation coefficient, that each of the components brings to the Met-
Table 6: Metrics correlation with human judgment on system outputs from the NLG2017 challenge. Here, $r$: Spearman’s correlation co-efficient, $\rho$: Spearman’s correlation co-efficient, $\tau$: Kendall’s Tau.

<table>
<thead>
<tr>
<th>Perturbation methods</th>
<th>BLEU $\rho$</th>
<th>METEOR $\rho$</th>
<th>BERTScore $\rho$</th>
<th>MoverScore $\rho$</th>
<th>RoMe $\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity replacement</td>
<td>0.06</td>
<td>0.06</td>
<td>0.08</td>
<td>0.09</td>
<td>0.14</td>
</tr>
<tr>
<td>Adjective replacement</td>
<td>0.07</td>
<td>0.07</td>
<td>0.13</td>
<td>0.11</td>
<td>0.14</td>
</tr>
<tr>
<td>Random word replacement</td>
<td>0.05</td>
<td>0.06</td>
<td>0.06</td>
<td>0.08</td>
<td>0.15</td>
</tr>
<tr>
<td>Text transformation</td>
<td>0.03</td>
<td>0.03</td>
<td>0.08</td>
<td>0.09</td>
<td>0.15</td>
</tr>
<tr>
<td>Passive form</td>
<td>0.02</td>
<td>0.04</td>
<td>0.08</td>
<td>0.10</td>
<td>0.21</td>
</tr>
</tbody>
</table>

Table 7: Metrics Spearman correlation score against human judgment on perturbed texts. Here, $f$: fluency, $s$: semantic similarity, $g$: grammatical correctness.

Table 8: Ablation study to understand such cases better.

Case 1: Entity Replacement. We perform Invariance test (INV) from (Ribeiro et al., 2020) to check the metrics’ NER capability in the text quality assessment. In this approach, we replace the entities present in the text partially or fully with other entities in the dataset. For instance, "The population of Germany" gets transformed to "The population of England".

Case 2: Adjective Replacement. Similar to the entity replacement, in this case we choose 100 data from the KELM dataset that contain adjective in them. Then we replace the adjectives with a synonym and an antonym word to generate two sentences from a single data. For instance, the adjective different is replaced with unlike and same. At the end of this process, we obtain 200 data.
Case 3: Random word replacement. We replace words in different positions in the text, with a generic token "AAA" following the adversarial text attack method from (Morris et al., 2020). For instance, the sentence, "x is a cheap restaurant near y" is transformed into "x is a cheap restaurant AAA AAA". We select the greedy search method with the constraints on stop-words modification from the TextAttack tool. This approach generates repetitive words in the text.

Case 4: Text transformation. We leverage TextFooler (Jin et al., 2020) to replace two words in the texts by similar words, keeping the semantic meaning and grammar preserved.

Case 5: Passive Forms. In this case, we randomly choose 200 data from the KELM (Agarwal et al., 2021) dataset where the system generated responses are in passive form.

From the results of robustness analysis in Table 7, it is evident that almost all the metrics obtain very low correlation scores with respect to human judgment. Word-overlap based metrics such as BLEU and METEOR mostly suffer from it. Although, RoMe achieves higher correlation scores in most of the cases, there is still scope for improvement in handling the fluency of the text better. Text perturbation techniques used to design the test cases often generate disfluent texts. In some cases, the texts’ entities or subjects get replaced by words from out of the domain. From our observation, we hypothesize that handling keywords such as entities can lead to a better correlation score.

5 Related Work

A potentially good evaluation metric is one that correlates highly with human judgment. Among the unsupervised approaches, BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and ROUGE (Lin, 2004) are the most popular evaluation metrics traditionally used for evaluating NLG systems. Although these metrics perform well in evaluating machine translation (MT) and summarization tasks, (Liu et al., 2016) shows that none of the word overlap based metrics is close to human level performance in dialogue system evaluation scenarios. In a different line of work, word embedding based metrics are introduced for evaluating NLG systems (Mikolov et al., 2013; Matsuo et al., 2017). Several unsupervised automated metrics were proposed that leverage EMD; one of them is Word Mover’s distance (WMD) (Kusner et al., 2015b). Later, (Matsuo et al., 2017) proposed an evaluation metric, incorporating WMD and word-embedding, where they use word alignment between the reference and hypothesis to handle the word-order problem. Recently, (Echizen-ya et al., 2019) introduced an EMD-based metric WE_WPI that utilizes the word-position information to tackle the differences in surface syntax in reference and hypothesis.

Several supervised metrics were also proposed for evaluating NLG. ADEM (Lowe et al., 2017) uses a RNN-based network to predict the human evaluation scores. With the recent development of language model-based pre-trained models (Zhang et al., 2020) proposed BERTScore, which uses a pre-trained BERT model for evaluating various NLG tasks such as machine translation and image captions. Recently, (Zhao et al., 2019) proposed MoverScore, which utilizes contextualized embedding to compute the mover’s score on word and sentence level. A notable difference between MoverScore and BERTScore is that the latter relies on hard alignment compared to soft alignments in the former. Unlike the previous methods, RoMe focuses on handling the sentence’s word repetition and passive form when computing the EMD score. Furthermore, RoMe trains a classifier by considering the sentence’s semantic, syntactic, and grammatical acceptability features to generate the final evaluation score.

6 Conclusion

We have presented RoMe, an automatic and robust evaluation metric for evaluating a variety of NLG tasks. The key contributions of RoMe include

1) **EMD-based semantic similarity**, where *hard word alignment and soft-penalization* techniques are employed into the EMD for tackling repetitive words and passive form of the sentence,

2) **Semantically enhanced TED** that computes the syntactic similarity based on the node-similarity of the parsed dependency trees,

3) **Grammatical acceptability classifier**, which evaluates the text’s grammatical quality, and

4) **Robustness analysis**, which assess the metric’s capability of handling various form of the text. Both quantitative and qualitative analyses exhibit that RoMe highly correlates with human judgment. We intend to extend RoMe by including more languages in the future.
References


M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger. 2015a. From word embeddings to document distances. In ICML.


A Appendix

A.1 Dependency Tree representation for Tree Edit Distance calculation

This section describes the process of preparing a dependency tree from a sentence, for computing TED-SE. Let us consider a reference statement "the aidaluna is operated by aida cruises which are located at rostock." and a hypothesis, "aida cruises, which is in rostock, operates aidaluna.". First, a dependency tree is parsed utilizing the Stanza dependency parser (Qi et al., 2020) and then converted to an adjacency list. The adjacency list contains a key-value pair oriented data structure where each key corresponds to a node’s index in the tree, and the value is a list of edges on which the head node is incident. Figure 6 shows the dependency trees and their corresponding adjacency lists for the given reference and hypothesis. List of nodes and adjacency lists are then fed into the TED-SE algorithm to calculate semantically enhanced tree edit distance as described in §3.3.

A.2 Model Parameters: Grammatical acceptability classifier

Table 8 shows the parameters used in BERT-base (Devlin et al., 2019) based binary classifier for computing grammatical acceptability (discussed in §3.3).

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td># of attention heads</td>
<td>12</td>
</tr>
<tr>
<td># of hidden layers</td>
<td>12</td>
</tr>
<tr>
<td>Hidden size</td>
<td>768</td>
</tr>
<tr>
<td>Hidden layer dropout</td>
<td>0.1</td>
</tr>
<tr>
<td>Layer norm epsilon</td>
<td>1e-12</td>
</tr>
<tr>
<td>Maximum positional embedding</td>
<td>512</td>
</tr>
<tr>
<td>Activation function</td>
<td>GELU</td>
</tr>
</tbody>
</table>

Table 8: Training parameters.

A.4 Hyper-parameter Settings

We use $\delta = 0.6$ and $\theta = 0.65$ in §3.1. Best values are found by a hyper-parameter search from a range of $[0,1.0]$ with an interval of 0.1. RoMe obtained the best result by utilizing ALBERT-large (Lan et al., 2020) model with 18M parameters and 24 layers. Furthermore, we use the English word embedding of dimension 300 to obtain results from Fasttext (Bojanowski et al., 2017) throughout the paper. We use a single GPU with 12GBs of memory for all the evaluations.

A.3 Annotation Tool

For all the annotation processes, we use the annotation tool shown in Figure 7. The tool is developed using Python programming language. Annotators can load their data into the tool in JSON format by selecting the Load Raw Data button. An example annotation step is shown in Figure 7. The reference and hypothesis sentences are displayed in different text windows. The annotators were asked to annotate the data based on Fluency, Semantically correctness and Grammar. Annotators can choose a value on a scale of $[1,3]$ for each category, from the corresponding drop-down option. Finally, the annotated text can be saved for evaluation using the save button, which saves the annotated data in JSON format.
**Ref:** the aidaluna is operated by aida cruises which are located at rostock.

**Dependency tree:**

```
  not  
     / \ 
  aux-pass aux-pass
    /     \ 
aid  aida cruises which are located at rostock.
  
the  
   / \ 
aid
```

**Adjacency list:** [0: [], 1: [0], 2: [], 3: [1, 2, 6], 4: [], 5: [], 6: [4, 5, 9], 7: [], 8: [], 9: [7, 8, 11], 10: [], 11: [10]]

**Nodes:** ['the', 'aidaluna', 'be', 'operate', 'by', 'aida', 'cruise', 'which', 'be', 'locate', 'at', 'rostock']

**Ref-tree (lemmas):** operate(aidaluna(the), be, cruise(by, aida, locate(which, be, rostock(at))))

**Hyp:** aida cruises, which is in rostock, operates aidaluna.

**Dependency tree:**

```
  not  
     / \ 
  aux-pass aux-pass
    /     \ 
adia cruises which is in rostock operates aidaluna.
```

**Adjacency list:** [0: [], 1: [0, 5], 2: [], 3: [4, []], 4: [1, 7], 5: [2, 3, 4], 6: [1, 7], 7: []]

**Nodes:** ['aida', 'cruise', 'which', 'be', 'in', 'rostock', 'operate', 'aidaluna']

**Hyp-tree (lemmas):** operate(cruise(aida, rostock(which, be, in)), aidaluna)

Figure 6: Dependency trees for reference and hypothesis, pre-processed for the TED calculation.

---

**Figure 7:** The annotation tool used by the annotators.