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 acceptability of the generated hypotheses. **Finally**, the scores obtained from the preceding steps are combined to form a representation vector, which is subsequently fed into a self-supervised network. The network produces a final score, referred to as RoMe's output which represents the overall quality of the hypothesis statement.

We investigate the effectiveness of our proposed metric by conducting experiments on datasets from various domains of NLG such as knowledge graph based language generation dataset (KELM (Agarwal et al., 2021)), dialogue datasets (Eric et al., 2017; Chaudhuri et al., 2021), the NLG2017 challenge dataset (Shimorina et al., 2018), structured data to language generation dataset (BAGEL (Mairesse et al., 2010) and SFHOTEL (Wen et al., 2015)). The capability of existing metrics to handle various forms of text has lately become a matter of debate in the NLP community (Ribeiro et al., 2020; Novikova et al., 2017; Liu et al., 2016). Hence, we conduct an extensive robustness analysis to assess RoMe's performance in handling diverse forms of system-generated sentences. To verify our claim, we design the analysis based on the text perturbation methods used in CHECKLIST (Ribeiro et al., 2020) and adversarial text transformation techniques from TextFooler (Jin et al., 2020) and TextAttack (Morris et al., 2020). Empirical assessment on benchmark datasets and the robustness analysis results exhibit that RoMe can handle various surface forms and generate an evaluation score, which highly correlates with human judgment. RoMe is designed to function at the sentence level and can be used to evaluate English sentences in the current version of the implementation. In the future versions, we plan to extend RoMe by including more languages. We released the code and the annotation tool that we used to get human annotation public [1].

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 $\mathcal{R} = \{r_1, r_2, ..., r_p\}$ and the hypothesis as $\mathcal{H} = \{h_1, h_2, ..., h_q\}$, 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:

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

where d_{ij} is the distance between the words r_i and h_j in the space and $\mathcal{F}(\mathcal{H}, \mathcal{R})$ 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:

$$\begin{aligned} & f_{ij} \geq 0 \qquad i = 1, 2, ..., p \text{ and } j = 1, 2, .., q, \\ & \sum_{j=1}^{q} f_{ij} \leq m_i \qquad i = 1, 2, ..., p, \\ & \sum_{i=1}^{p} f_{ij} \leq n_j \qquad j = 1, 2, ..., q, \\ & \sum_{i=1}^{p} \sum_{j=1}^{q} f_{ij} = \min(m_{\sum}, n_{\sum}) \end{aligned} \quad (2)$$

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

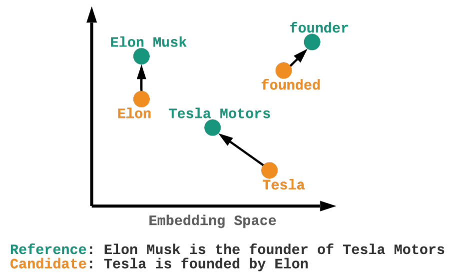

Figure 1: Illustrating an abstraction of the EMD.

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.

[1] The URL of the Github repository will appear here.

Formally, a dependency tree is a set of nodes $\Omega = \{w_0, w_1, ..., w_k\}$ and a set of dependency links $\mathcal{G} = \{g_0, g_1, ..., g_k\}$, where w_0 is the imaginary root node and g_i is an index into Ω representing the governor of w_i. Every node has exactly one governor except for w_0, which has no governor (Hall and Novák, 2010). Syntactic similarity between a pair of dependency trees can be estimated using several methods, such as graph centralities 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 hypothesis. 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*.

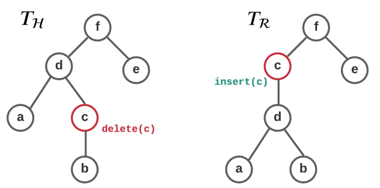

Figure 2: Visualization of the required *edit* operations to transform $T_{\mathcal{H}}$ to $T_{\mathcal{R}}$. The operations corresponds to the following sequence: *delete*(node with label c), *insert*(node with label c).

Lets consider $T_{\mathcal{H}}$ and $T_{\mathcal{H}}$ be the parsed dependency trees of the hypothesis and reference, respectively. 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 similarity 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 *Semantic Word Alignment* and *Soft-penalization* techniques 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 similarities 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 modeling inter-sentence coherence, which improves multi-sentence encoding tasks.

The word alignment score is computed as follows:

$$\mathcal{A}(r_i, h_j) = \frac{\vec{r_i} \cdot \vec{h_j}}{\|\vec{r_i}\| \|\vec{h_j}\|} \cdot \frac{|q\,(i+1) - p\,(j+1)\,|}{pq} \qquad (3)$$

where $\vec{r_i}$ and $\vec{h_j}$ 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 $\vec{r_i}$ and $\vec{h_j}$, and the second part calculates the relative position information as proposed in (Echizen-ya et al., 2019).

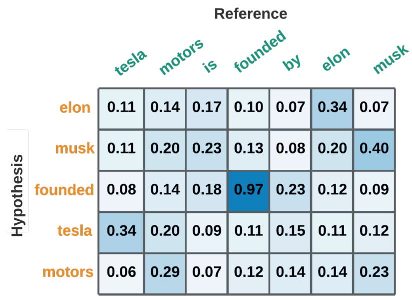

Figure 3: An example word alignment matrix for the reference sentence: "*tesla motors is founded by elon musk*" and its passive form: "*elon musk founded tesla motors*" is illustrated here.

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 Figure 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 hypothesis from getting aligned to multiple words in the reference as illustrated by the dotted arrows in Figure 4. We denote the resulting set of hard-aligned word pairs as $\mathcal{A}_{hc}$.

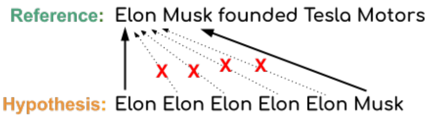

Figure 4: An adversarial example of a hypothesis containing repetitive words.

Transport Distance. A distance matrix $\mathcal{D}$ is required to compute the final EMD score. For each aligned pair $(r_i, h_j) \in \mathcal{A}_{hc}$, where $\frac{\vec{r_i} \cdot \vec{h_j}}{\|\vec{r_i}\| \|\vec{h_j}\|} > \delta$, the distance between r_i and h_j is computed as follows:

$$d_{ij} = 1.0 - \frac{\vec{r_i} \cdot \vec{h_j}}{\|\vec{r_i}\| \|\vec{h_j}\|} \cdot e^{\gamma \cdot \frac{|q(i+1)-p(j+1)|}{pq}} \qquad (4)$$

where $d_{ij} \in \mathcal{D}$ and δ is a confidence threshold found via hyper-parameter serach, $\gamma \in [-1, 0]$ is a soft-penalization constant. For all the non-hard-aligned pairs and aligned pairs with value less than δ, the distance d_{ij} receives a maximum value of 1.0. Intuitively, a lower value of d_{ij} implies that the word needs to travel a shorter distance in the transportation problem of EMD. In Equation 4, $e^{\gamma \cdot \frac{|q(i+1)-p(j+1)|}{pq}}$ works as a penalty where a higher position difference multiplied with the negative constant γ will results in low d_{ij} score. The role of γ is explained below.

Soft-penalization. Existing metrics often impose hard penalties for words with different order than the reference sentence (Zhao et al., 2019; Echizen-ya et al., 2019). For instance, sentences phrased in the passive form obtain a very low score in those metrics. Addressing this issue, we introduce a soft-penalization constant $\gamma = -\frac{|j-i|}{\max(p,q)}$ in Equation 4 to handle the passive form of a sentence better. Lets consider a reference, "*Shakespeare has written Macbeth*" and the passive form of the sentence as hypothesis, "*The Macbeth is written by Shakespeare*". The word *Shakespeare* appears at the beginning of the reference and at the end of the hypothesis, thus the position difference is larger. To the words in such scenario, γ imposes a lower penalty as it divides the position difference by the length $max(p, q)$.

Finally, following the optimization constraints of Equation 2 we obtain the transportation flow $\mathcal{F}(\mathcal{H}, \mathcal{R})$. For the optimized flow $f_{ij} \in \mathcal{F}(\mathcal{H}, \mathcal{R})$, the final equation of EMD is as follows:

$$EMD(\mathcal{H}, \mathcal{R}) = \frac{\min_{f_{ij} \in \mathcal{F}(\mathcal{H}, \mathcal{R})} \sum_{i=1}^{p} \sum_{j=1}^{q} d_{ij} f_{ij}}{\min(m_{\sum}, n_{\sum})} \qquad (5)$$

The semantic similarity between hypothesis and reference is denoted as $\mathcal{F}_{sem} = 1.0 - EMD$. The normalized value of EMD is used to calculate $\mathcal{F}_{sem}$.

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 θThis 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 $\mathcal{F}_{ted}$. 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 utilize 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 donote the score from the classifier as the feature $\mathcal{F}_g$, which is used to train a neural network (see §3.4). Model parameters can be found in Appendix A.2.

Settings	Metrics	BAGEL			SFHOTEL		
		Info	Nat	Qual	Info	Nat	Qual
Baselines	BLEU-1	0.225	0.141	0.113	0.107	0.175	0.069
	BLEU-2	0.211	0.152	0.115	0.097	0.174	0.071
	METEOR	0.251	0.127	0.116	0.163	0.193	0.118
	BERTScore	0.267	0.210	0.178	0.163	0.193	0.118
	SMD+W2V	0.024	0.074	0.078	0.022	0.025	0.011
	SMD+ELMO+PMEANS	0.251	0.171	0.147	0.130	0.176	0.096
	SMD+BERT+MNLI+PMAENS	0.280	0.149	0.120	0.205	0.239	0.147
	WMD-1+ELMO+PMEANS	0.261	0.163	0.148	0.147	0.215	0.136
	WMD-1+BERT+PMEANS	0.298	0.212	0.163	0.203	0.261	0.182
	WMD-1+BERT+MNLI+PMEANS	0.285	0.195	0.158	0.207	0.270	0.183
RoMe	RoMe (Fasttext)	0.112	0.163	0.132	0.172	0.190	0.231
	RoMe (BERT)	0.160	0.251	0.202	0.212	0.283	0.300
	RoMe (ALBERT-base)	0.162	0.259	0.222	0.231	0.295	0.315
	RoMe (ALBERT-large)	0.170	0.274	0.241	0.244	0.320	0.327

Table 1: Spearman correlation (ρ) scores computed from the metric scores with respect to the human evaluation scores for BAGEL and SFHOTEL datasets. Baseline model's results are reported form (Zhao et al., 2019). Here, **Info**, **Nat** and **Qual** referes to *informativeness*, *naturalness*, and *quality*, respectively.

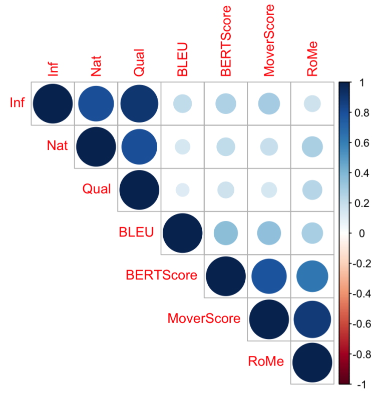

Figure 5: Correlation between the explored metrics.

3.4 Final Scorer Network

A feed-forward neural network takes the previously computed features as input and learns a function $f(\mathcal{F}_{sem};\mathcal{F}_{ted};\mathcal{F}_g)$ 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: *Fluency*, *Semantic Correctness*, and *Grammatical correctness*. In this case, the values stand for 1: *poor*, 2: *average*, and 3: *good*. The overall inter-annotator agreement score, κ 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-

	Text	EMD	TED-SE	Grammar	RoMe
$\mathcal{R}$ $\mathcal{H}$	Munich is located at the southern part of Germany. Munich is situated in the south of Germany.	0.83	1.0	0.94	0.80
$\mathcal{R}$ $\mathcal{H}$	Tesla motors is founded by Elon Musk. Elon Musk has founded Tesla Motors.	0.70	0.85	0.96	0.69
$\mathcal{R}$ $\mathcal{H}$	Elon musk has founded tesla motors. Elon elon elon elon elon founded tesla tesla tesla.	0.01	0.50	0.17	0.11

Table 2: Component-wise qualitative analysis.

	Text	BLEU	BERTScore	MoverScore	RoMe
$\mathcal{R}$ $\mathcal{H}$	James Craig Watson, who died from peritonitis, discovered 101 Helena. The Polish Academy of Science is regionserved.	0.0	0.81	0.54	0.15
$\mathcal{R}$ $\mathcal{H}$	1001 gaussia was formerly known as 1923 oaa907 xc. The former name for the former name for 11 gunger is 1923. One of the former name is 1923.	0.0	0.79	0.51	0.13

Table 3: Qualitative analysis.

Dialogue dataset	Models	correlation	SentBLEU	METEOR	BERTScore	MoverScore	RoMe
In-car dialogue	Mem2Seq	ρ	0.07	0.35	0.40	0.49	0.51
	GLMP	ρ	0.04	0.29	0.32	0.31	0.32
	DialoGPT	ρ	0.17	0.60	0.62	0.73	0.78
Soccer dialogue	Mem2Seq	ρ	0.03	0.08	0.08	0.11	0.11
	GLMP	ρ	0.02	0.08	0.03	0.12	0.14
	DialoGPT	ρ	0.04	0.26	0.31	0.39	0.43

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

Approaches	Correlation (ρ)
RoMe with EMD$_{std}$	64.8
+ EMD$_{align}$	66.0
+ EMD$_{soft}$	66.9
+ TED-SE	69.1
+ Grammar	70.1

Table 5: Ablation Study.

Score (Zhao et al., 2019). We evaluate the metrics on the sentence level to make a fair comparison.

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-

Metrics	BLEU			METEOR			BERTScore			MoverScore			RoMe		
Systems	ρ	r	τ	ρ	r	τ	ρ	r	τ	ρ	r	τ	ρ	r	τ
ADAPT	0.38	0.39	0.27	0.57	0.58	0.41	0.61	0.72	0.50	0.68	0.73	0.49	0.72	0.70	0.51
Baseline	0.35	0.42	0.26	0.49	0.49	0.33	0.49	0.50	0.35	0.59	0.61	0.43	0.53	0.53	0.37
melbourne	0.32	0.31	0.21	0.35	0.35	0.24	0.33	0.33	0.26	0.40	0.39	0.28	0.44	0.50	0.35
Pkuwriter	0.37	0.38	0.28	0.47	0.47	0.31	0.48	0.53	0.38	0.57	0.56	0.39	0.58	0.56	0.39
tilburg-nmt	0.25	0.20	0.13	0.26	0.26	0.18	0.38	0.39	0.30	0.49	0.50	0.36	0.64	0.68	0.50
tilburg-pipe	0.38	0.41	0.30	0.52	0.43	0.30	0.53	0.48	0.33	0.62	0.50	0.35	0.38	0.42	0.27
tilburg-smt	0.25	0.20	0.13	0.21	0.19	0.13	0.33	0.30	0.25	0.40	0.38	0.27	0.50	0.51	0.36
upf-forge	0.14	0.13	0.08	0.13	0.11	0.08	0.26	0.25	0.19	0.27	0.27	0.18	0.42	0.42	0.30
vietnam	0.73	0.80	0.62	0.87	0.90	0.72	0.81	0.76	0.70	0.90	0.78	0.73	0.84	0.89	0.83

Table 6: Metrics correlation with human judgment on system outputs from the NLG2017 challenge. Here, r: Pearson correlation co-efficient, ρ: Spearman's correlation co-efficient, τ: Kendall's Tau.

Metrics	BLEU			METEOR			BERTScore			MoverScore			RoMe		
Perturbation methods	f	s	g	f	s	g	f	s	g	f	s	g	f	s	g
Entity replacement	0.06	0.04	0.06	0.09	0.09	0.08	0.11	0.07	0.09	0.16	0.13	0.11	0.16	0.19	0.14
Adjective replacement	0.07	0.06	0.07	0.09	0.13	0.11	0.11	0.11	0.13	0.13	0.17	0.16	0.18	0.23	0.18
Random word replacement	0.05	0.06	0.03	0.06	0.06	0.05	0.11	0.10	0.08	0.11	0.13	0.09	0.15	0.15	0.23
Text transformation	0.03	0.01	0.03	0.08	0.09	0.07	0.13	0.15	0.15	0.15	0.18	0.19	0.18	0.19	0.21
Passive form	0.02	0.01	0.04	0.08	0.10	0.08	0.19	0.24	0.21	0.23	0.24	0.22	0.25	0.28	0.28

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

ric. We randomly choose 100 system-generated dialogue utterances from the datasets used in Table 4, since they contain sentences in passive form and repetitive words. The correlation of standard EMD with the human judgement is denoted as "RoMe score with EMD$_{std}$". Inclusion of semantic word alignment (EMD$_{align}$) and soft-penalization (EMD$_{soft}$) further improved the correlation score. Till this part of the ablation, the EMD score was directly compared to human judgement without the use of neural classifier to obtain the correlation score. Moreover, the correlation score improved significantly when the semantically enhanced TED and grammatical acceptability were introduced as features in addition to the EMD score to a neural classifier. We hypothesize that the inclusion of language features related to grammar and syntactic similarity helped the neural network achieve better performance.

4.5 Qualitative Analysis

RoMe is developed in a modular fashion, so it may be used to generate scores for semantic similarity, syntactic similarity, and grammatical acceptability separately. Table 2 shows the component-wise score and the final score of RoMe on three example data. In the first example, RoMe demonstrates its ability of capturing similar sentences by obtaining high score. The scores from several components in the second example demonstrate RoMe's ability to handle passive form. The final example in Table 2 demonstrates that RoMe penalizes sentence with repetitive word.

Table 3 shows the performance of the three baselines and RoMe in handling erroneous cases. Although the first example contain completely different hypothesis and the second case with repetitive hypothesis both BERTScore and MoverScore exhibit high score. On the contrary, BLEU score is unable to handle such scenarios. However, by obtaining low scores, RoMe demonstrates its ability to understand such cases better.

4.6 Robustness Analysis

In this section, we design five test cases to stress the models capability. For the analysis purpose, we randomly sample data from KELM (Agarwal et al., 2021) and BAGEL (Mairesse et al., 2010). Then, the annotators annotate the sampled data on the following criteria: *fluency*, *semantically correctness*, *grammatically correctness*.

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 are 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

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, Online. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Debanjan Chaudhuri, Md Rashad Al Hasan Rony, Simon Jordan, and Jens Lehmann. 2019. Using a kg-copy network for non-goal oriented dialogues. In *International Semantic Web Conference*, pages 93–109. Springer.

Debanjan Chaudhuri, Md Rashad Al Hasan Rony, and Jens Lehmann. 2021. Grounding dialogue systems via knowledge graph aware decoding with pre-trained transformers. In *Eighteenth Extended Semantic Web Conference - Research Track*.

Robert Dale and Chris Mellish. 1998. Towards evaluation in natural language generation. In *In Proceedings of First International Conference on Language Resources and Evaluation*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hiroshi Echizen-ya, Kenji Araki, and Eduard Hovy. 2019. Word embedding-based automatic mt evaluation metric using word position information. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1874–1883.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany. Association for Computational Linguistics.

Keith Hall and Václav Novák. 2010. Corrective dependency parsing. *Trends in Parsing Technology*, pages 151–167.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

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

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015b. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966.

Andrey Kutuzov and Elizaveta Kuzmenko. 2019. To lemmatize or not to lemmatize: how word normalisation affects elmo performance in word sense disambiguation. *arXiv preprint arXiv:1909.03135*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic Turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1116–1126, Vancouver, Canada. Association for Computational Linguistics.

François Mairesse, Milica Gašić, Filip Jurčíček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2010. Phrase-based statistical language generation using graphical models and active learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, page 1552–1561, USA. Association for Computational Linguistics.

Junki Matsuo, Mamoru Komachi, and Katsuhito Sudoh. 2017. Word-alignment-based segment-level machine translation evaluation using word embeddings. *arXiv preprint arXiv:1704.00380*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. Association for Computational Linguistics.

Masanori Oya. 2020. Syntactic similarity of the sentences in a multi-lingual parallel corpus based on the euclidean distance of their dependency trees. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 225–233.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 1998. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 59–66. IEEE.

Anastasia Shimorina, Claire Gardent, Shashi Narayan, and Laura Perez-Beltrachini. 2018. WebNLG Challenge: Human Evaluation Results. Technical report, Loria & Inria Grand Est.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Kaizhong Zhang and Dennis Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

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).

Parameters	**Values**
# of attention heads	12
# of hidden layers	12
Hidden size	768
Hidden layer dropout	0.1
Layer norm epsilon	1e-12
Maximum positional embedding	512
Activation function	GELU

Table 8: Training parameters.

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.

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 singe GPU with 12GBs of memory for all the evaluations.

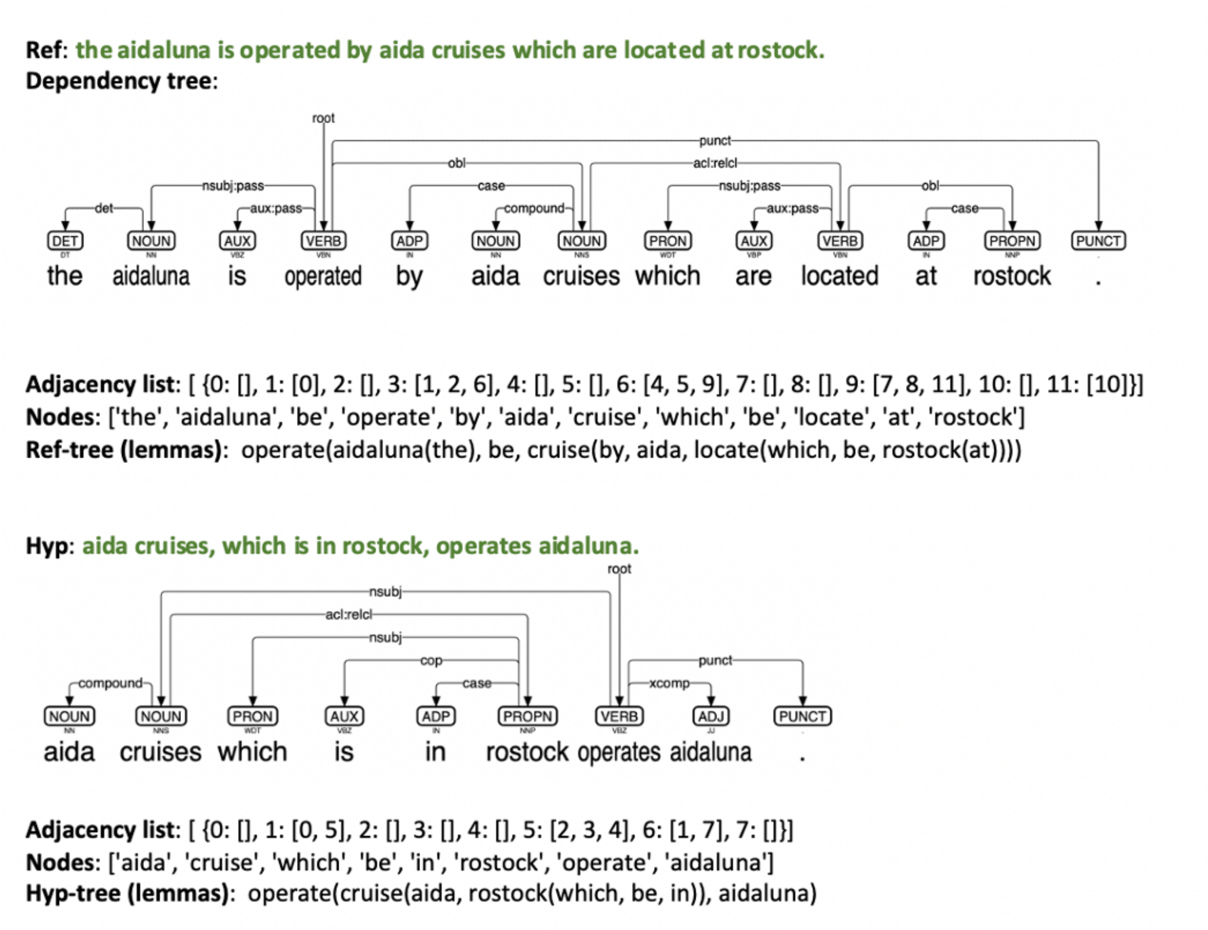

Figure 6: Dependency trees for reference and hypothesis, pre-processed for the TED calculation.

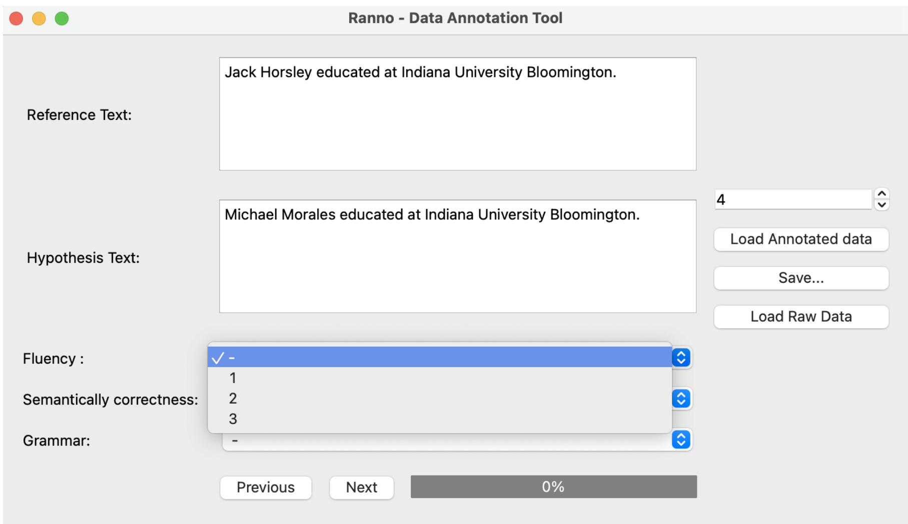

Figure 7: The annotation tool used by the annotators.