ARGENT: Automatic Reference-free Evaluation for Open-Ended Text Generation without Source Inputs

Xinyue Zhang*[1], Agathe Zecevic*[2,3], Sebastian Zeki[2], Angus Roberts[1]

[1]Biostatistics and Health Informatics, Institute of Psychiatry, Psychology and Neuroscience
King's College London, United Kingdom,
[2]Gastroenterology Department, Guy's and St Thomas' NHS Foundation Trust, United Kingdom,
[3]Clinical Scientific Computing, Guy's and St Thomas' NHS Foundation Trust, United Kingdom

Correspondence: leo.xinyue.zhang@kcl.ac.uk, agathe.zecevic@gstt.nhs.uk, **Joint first authorship*

Abstract

With increased accessibility of machine-generated texts, the need for their evaluation has also grown. There are broadly two types of text generation tasks. In open-ended generation tasks (OGTs), the model generates de novo text without any input on which to base it, such as story generation. In reflective generation tasks (RGTs), the model output is generated to reflect an input sequence, such as in machine translation. There are many studies on RGT evaluation, where the metrics typically compare one or more gold-standard references to the model output. Evaluation of OGTs has received less attention and is more challenging: since the task does not aim to reflect an input, there are usually no reference texts. In this paper, we propose a new perspective that unifies OGT evaluation with RGT evaluation, based on which we develop an automatic, reference-free generative text evaluation model (ARGENT), and review previous literature from this perspective. Our experiments demonstrate the effectiveness of these methods across informal, formal, and domain-specific texts. We conduct a meta-evaluation to compare existing and proposed metrics, finding that our approach aligns more closely with human judgement.

1 Introduction

Natural language generation (NLG) has progressed significantly in the last decade. This progress has been made through the use of encoder-decoder (Lewis et al., 2020) and decoder only architectures (Brown et al., 2020; Touvron et al., 2023). In the last few years, the use of these transformer-based architectures (Vaswani et al., 2017) and increased compute capacity to create generative Large Language Models (LLMs) such as Brown et al. (2020); Touvron et al. (2023) has attracted attention from both academia and the public. However, the lack of robust evaluation metrics for generated text has limited the ability to make informed choices among candidate outputs produced by one or more LLMs.

NLG tasks can be categorised on a spectrum between two categories: reflective generation tasks (RTGs)[1] and open-ended generation tasks (OTGs). In RGTs, the output closely reflects the content of the input and must remain faithful to it, such as machine translation and summarisation. OGTs, by contrast, involve generating novel content that is not directly grounded in the input, such as story generation or synthetic medical report creation. Rather than a strict dichotomy, generation tasks are better understood as positioning on a spectrum of constraint. For example, image captioning and text expansion lie between highly constrained tasks such as translation and unconstrained tasks such as storytelling.

Many studies on RGTs, such as machine translation and summarisation, evaluate output quality by comparing model-generated texts to one or more pre-written human references, using similarity metrics such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), BEER (Stanojević and Sima'an, 2014), BERTScore (Zhang et al., 2020), BLEURT (Sellam et al., 2020), and COMET (Rei et al., 2020a). However, these approaches often depend heavily on reference selection, which can significantly impact evaluation outcomes. More recent work on quality estimation (QE), such as COMET-QE (Rei et al., 2020b), addresses this issue by evaluating outputs in relation to source inputs without requiring human references (Zhao et al., 2024). While this mitigates the problem of reference selection, it remains applicable only to RGTs, as it still relies on source inputs. In contrast, open-ended OGTs, such as story

[1] We use the term "reflective generation" to emphasise the output is semantically grounded in an input. While this may sometimes align with what is commonly called "task-oriented generation". We adopt this term to contrast explicitly with open-ended generation.

Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM² 2025), pages 82–98
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

or dialogue generation, remain under-explored in this context, largely due to the difficulty of defining appropriate references for outputs that are not input-grounded (Yue et al., 2023). As a result, OGT evaluation often relies on *distribution-level* comparisons between model-generated and human-written corpora in the target domain. Common approaches include statistical metrics such as self-BLEU (Zhu et al., 2018) and generation perplexity (Bhandari et al., 2020), as well as divergence-based techniques such as Mauve (Pillutla et al., 2021), which estimates the difference between synthetic and human text distributions using Kullback-Leibler (KL) divergence.

These evaluation methods have two major problems: (1) in OGT evaluation, they are unable to assess the quality of each individual output; (2) There is no unified conceptual framework for comparing metrics across RGT and OGT paradigms. This limits the transfer of insights and tools between these domains, especially transferring tools from RGT to OGT.

This paper addresses these issues by proposing a unified evaluation framework that bridges RGT and OGT evaluation. Within this framework, we introduce a new reference-free method for evaluating OGTs without source inputs at the level of individual outputs, which we call **ARGENT** (**A**utomatic **R**eference-free **GEN**erated **T**ext evaluation). To benchmark ARGENT, we also develop a meta-evaluation framework to assess the effectiveness of evaluation metrics themselves.

The contributions of this paper are as follows:

- We present a conceptual framework that connects evaluation practices across OGTs and RGTs.
- We propose ARGENT, a reference-free method for evaluating open-ended generation via corrupted text, and demonstrate that it performs competitively with or better than existing reference-based and reference-free baselines across informal, formal, and domain-specific tasks.
- We develop a scalable text corruption pipeline using inflection and shuffling techniques to simulate a range of quality variations.
- We introduce a meta-evaluation framework for assessing evaluation metrics without requiring human labels.

2 Bridging OGT with RGT evaluation from a unified framework

Evaluating language generation differs fundamentally from evaluating traditional classification or regression tasks. In classification, there exists a finite list of output classes; in regression, outputs lie on a continuous and measurable scale. In contrast, most language generation tasks do not have a single correct answer, and many do not even have a finite set of acceptable answers. Instead, evaluation typically relies on a set of human-written references. Moreover, language generation lacks an inherent numerical ground truth, which requires the use of similarity functions to compare generated text to references.

We illustrate this complexity in Appendix A with a simple translation example to demonstrate how evaluation outcomes vary depending on (1) the references selected, and (2) the similarity function used.

In any evaluation of a text generation model, we can identify the following components:

- **Output** - the text generated by the model, e.g. candidate translation.
- **Reference space** - A set of all possible gold-standard references or correct outputs for the task, e.g. all valid translations of a given sentence, all valid summaries of a document.
- **Reference** - A single instance drawn from the reference space, often used as the "gold standard" for comparison.
- **Similarity score** - A function that measures similarity between the model output and a reference, such as BLEU, BERTScore, BLUERT, COMET.
- **Optimal reference** - The reference that is most similar to the model output according to the similarity function.

Let $\mathbf{Y}$ denote the set of all possible references, $\hat{Y}$ the output of the model, and $f_{similarity}$ the similarity score function. The evaluation score E for output $\hat{Y}$ is defined as:

$$E = max(f_{similarity}(\hat{Y}, Y_i), \forall Y_i \in \mathbf{Y}) \qquad (1)$$

The corresponding optimal reference, which depends on both the model output and the chosen similarity function, is defined as:

$$Y_{optimal}(\hat{Y}, f_{similarity}) = argmax(f_{similarity}(\hat{Y}, Y_i), \forall Y_i \in \mathbf{Y}) \qquad (2)$$

Key points arising from this formulation include:

- In the literature, the evaluation process and the similarity function are often conflated. However, the effectiveness of an evaluation depends on both the similarity function and the references used. In this paper, we define evaluation as the combination of reference selection and the similarity function.
- For a given output, the evaluation depends on the best-matching reference within the reference space under the chosen similarity function. Thus, the measured score is the maximum over all possible similarity scores with individual references.
- Some similarity functions are more effective than others. Functions that consider syntax and semantics typically align more closely with human judgments than those relying only on lexical overlap.
- This framework applies to both reflective and open-ended generation. The main difference lies in the size and structure of the reference spaces: RGTs typically have a small, well-defined reference set, whereas OGTs have much larger and more diverse reference spaces.

3 Auto-Evaluation for Language Quality

The large reference space in OGT evaluation leads to a challenge: how can we identify the closest reference to a given model output? One solution is to use output-oriented human annotation, in which a human judge corrects errors in an output by making the minimum number of changes, to give an error-free text. This revised text can then serve as the closest reference, and the output-reference pair can be used for evaluation. This technique has been applied in in RGTs, such as machine translation, where it has been shown to gives scores more aligned with human judgement than pre-written references with a translation edit rate metric (Snover et al., 2006). However, such output-oriented evaluation is costly and does not scale. We could overcome this with an automatic evaluation, but auto-evaluation may itself vary in quality, with some methods providing results more aligned with human judgement than others. We therefore need to consider ways in which we might measure the quality of auto-evaluations.

The remainder of this paper discusses a new reference-free auto-evaluation method, ARGENT, and meta-evaluations of ARGENT and existing metrics under different dataset conditions. Figure 1 shows the relationships between evaluation, ideal evaluation, auto-evaluation methods ARGENT, and meta-evaluation presented in this paper.

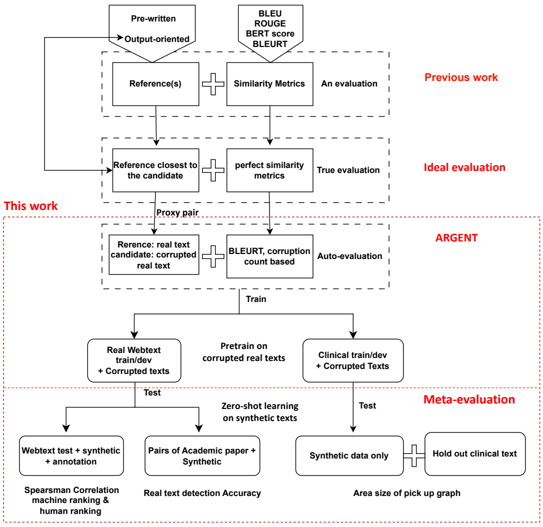

Pre-written
Output-oriented
BLEU
ROUGE
BERT score
BLEURT
Reference(s)
Similarity Metrics
An evaluation
Previous work
Reference closest to the candidate
perfect similarity metrics
True evaluation
Ideal evaluation
This work
Proxy pair
Rerence: real text candidate: corrupted real text
BLEURT, corruption count based
Auto-evaluation
ARGENT
Train
Real Webtext train/dev + Corrupted texts
Pretrain on corrupted real texts
Clinical train/dev + Corrupted Texts
Test
Zero-shot learning on synthetic texts
Test
Meta-evaluation
Webtext test + synthetic + annotation
Pairs of Academic paper + Synthetic
Synthetic data only
Hold out clinical text
Spearsman Correlation machine ranking & human ranking
Real text detection Accuracy
Area size of pick up graph

Figure 1: Relationships between different evaluation methods and experimental work presented in this paper

3.1 ARGENT : Pre-trained Auto-evaluation on Corrupted Texts

To understand automatic evaluation, consider Equation 2 as defining an ideal evaluation model. Given a set of all possible references and the output from a generative NLP model, this evaluation model would assign an evaluation score based on the highest similarity between the output and any valid reference. However, in practice, it is rarely feasible to enumerate the entire reference space and determine which reference yields the highest similarity score for a given output.

Suppose, however, that we could generate a set of proxy outputs, each associated with a known ideal evaluation score. We could then train a model to learn this mapping from output to the ideal evaluation score, effectively approximating the behaviour of the ideal evaluation model. Once trained, such a model would be able to predict the evaluation score for new, unseen outputs without requiring access to any references.

This is the intuition behind ARGENT. To create training data for ARGENT, we reverse the typical direction of evaluation. Instead of comparing an output to a reference, we start with a high-quality reference and apply controlled corruption strategies to simulate model-like outputs. These corrupted versions serve as proxy outputs, while the original, uncorrupted reference acts as the corresponding "ground truth" which is the closest reference to the corrupted proxy. By varying the degree of corruption, we can systematically control and quantify the quality of the proxy output relative to the reference. This gives us a diverse range of qualities of proxy outputs. ARGENT is then trained to predict these scores, allowing it to generalise to real model outputs and provide reference-free evaluation for generated texts.

Text corruption Text corruption methods need to reflect the variations in language quality in generated text. In this regard, we propose two text corruption methods, an inflection method and a local shuffling method.

In the inflection method, tokens in a sentence are inflected into different part-of-speech (POS) forms. For example, in the sentence "I like books," the token "books" is a plural noun. By inflecting it into the past-tense verb "booked", we obtain the corrupted sentence "I like booked." For POS tagging, we use the SpaCy tagger module[2], along with the `lemminflect` module[3] for inflection. As not all words can be inflected meaningfully, we restrict this process to tokens with POS tags in the following set: `JJ, JJR, JJS, NN, NNS, NNP, NNPS, RB, RBR, RBS, VB, VBD, VBG, VBN, VBP, VBZ`[1].

In the local shuffling method, we slide a window of variable length over the sentence and randomly shuffle the tokens within each window. The window size is sampled randomly from a predefined range. When both inflection and shuffling are applied to the same text, we refer to this process as shufflection.

The pseudo-code for both inflection and local shuffling applied to a single report can be found in Appendix B, Algorithms 1 and 2. To create a dataset with a range of quality levels, we vary the corruption rate for each report. Specifically, the corruption probabilities are sampled from a predefined range. The corresponding pseudo-code is provided in Appendix B, Algorithm 3.

We explore two methods for generating quality scores for corrupted output texts. The first method is based on the proportion of token-level changes made during corruption. Given a text of length N and K corruption steps, where the original (uncorrupted) token state is denoted as $k = 0$, the corruption score is defined as the proportion of altered tokens across all steps. The corresponding text quality score is computed as the complement of the corruption score:

$$S_{\text{corruption}} = \frac{1}{KN} \sum_{k=1}^{K} \sum_{i=1}^{N} \mathbb{1}(x_i^k \neq x_i^{k-1}) \quad (3)$$

$$S_{\text{quality}} = 1 - S_{\text{corruption}} \quad (4)$$

The second method uses BLEURT, a state-of-the-art evaluation metric originally developed for machine translation (RGT). (Sellam et al., 2020). BLEURT leverages contextual embeddings and is fine-tuned on human judgments to assess the semantic similarity between a reference and a candidate. In ARGENT, we use BLEURT to score each corrupted proxy output against its corresponding original (reference) text.

In both the corruption-count-based and BLEURT-based methods, the resulting score serves as the supervision signal for training the ARGENT model. That is, ARGENT learns to predict these scores from corrupted outputs without requiring access to references at inference time. By evaluating both scoring approaches, we explore ARGENT's sensitivity to different types of supervision signals, ranging from interpretable, token-level corruption counts to semantically-informed BLEURT scores. This comparison informs practical choices for similar reference-free evaluation tasks.

3.2 Meta-evaluation of evaluation models

For text generation datasets with human annotations, the correlation between automatic evaluation scores and human judgments is a common way to assess the performance of auto-evaluation models. However, obtaining consistent and reliable human annotations is difficult and often results in noisy or inconsistent labels (Clark et al., 2021; Karpinska et al., 2021). If the objective is to measure the language deviation of synthetic texts from real texts, it is reasonable to assume that the corresponding metrics of real texts should, on average, be no lower than that of synthetic ones. For example, in the case of synthetic clinical reports, their language is expected to deviate from the language used in real clinical reports. Based on this assumption, we propose the following two meta-evaluation techniques

[2] https://spacy.io/api/tagger

[3] https://spacy.io/universe/project/lemminflect

that do not rely on human annotation.

In some specific cases, datasets include pairs of real and semi-synthetic texts. For instance, Liyanage et al. (2022) construct such pairs by replacing a few sentences in real documents with generated ones, for use in synthetic text detection tasks. In such settings, auto-evaluation scores can be compared across each pair: a correct decision (true positive) is made when the real text receives a no lower score than its synthetic counterpart.

In scenarios where no such explicit pairs are available, we propose a batch-level evaluation approach. A batch of texts (e.g., 100 samples) is constructed containing a known mix of real and synthetic data, e.g. 90% synthetic and 10% real. The texts are then ranked according to their auto-evaluation scores. The top k% of ranked texts are then sampled, with k varying from 1 to 100. For each top k% (where k ranges from 1 to 100) subset, we calculate the percentage of real texts present in the subset. This quantity is referred to as the *pick-up rate*, i.e. the rate at which real texts are identified by the auto-evaluation model as high quality.

An example pick-up rate curve is shown in Figure 2, where the x axis represents the top k% of the ranked texts, and the y axis represents the percentage of real texts among those top k% (pick-up rate). For a 90% to 10% rate of synthetic to real texts, in the best case, all real texts appear in the top 10% of the ranking, forming the upper bound line. In the worst case, they appear in the bottom 10%, forming the lower bound. A random ranking would yield a diagonal line, where 10% of real texts are expected in every decile.

For an auto-evaluation model, the area between its curve and the lower bound reflects the quality of the auto-evaluation model. To quantify performance, we define a meta-evaluation score as the area between the model's pick-up rate curve and lower bound, normalised by the area between the upper and lower bounds. Since the score curve is discrete (from 0 to 100), the area is computed as the sum of vertical differences to the lower bound at each k. A random ranking diagonal line corresponds to 50% of the area between bounds, establishing a baseline score of 50%.

4 Experiments

Data and metrics: To evaluate our framework, we conducted experiments on three types of text: formal, informal, and domain-specific. We report results using three meta-evaluation criteria: correlation with human scores, pairwise accuracy, and the area under the pick-up rate curve. Details of the datasets and meta evaluations used for each type are provided in the corresponding subsections below.

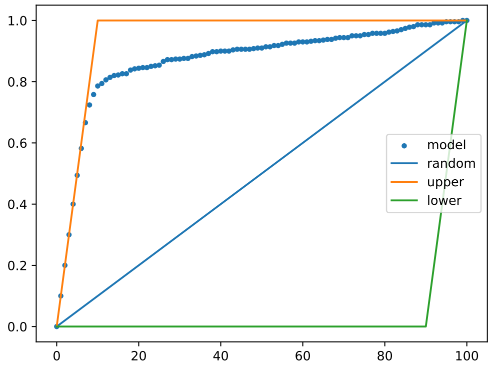


Figure 2: Example pick-up rate graph

Auto-evaluation models: Unless stated otherwise, all ARGENT auto-evaluation models reported in this paper are based on the BERT-base cased architecture (12 layers, 768 hidden units, 12 attention heads) (Devlin et al., 2019). ARGENT models are pre-trained on corrupted texts and applied directly to test tasks, consisting of either machine-generated or real texts, without fine-tuning on the test data. For pre-training, we use a batch size of 32, a learning rate of 1e-5, and train for 3 epochs. The model contains approximately 110 million parameters and was trained on a single NVIDIA A100 GPU.

Pre-training dataset: Unless stated otherwise, all pre-training datasets are constructed by applying inflection and local shuffling to real texts. We perform a grid search over inflection and shuffling probabilities in the range {0.2, 0.4, 0.6, 0.8, 1.0} for each corruption method. For shufflection, we use a pair of probability values, one for inflection and one for shuffling, that give the best performance for each method individually. Each corrupted text in the pre-training dataset is assigned a quality score using both the corruption-count-based method and the BLEURT-based method.

4.1 Informal Text Evaluation: WebText

Dataset and Metrics Evaluation on informal text is conducted using the WebText dataset.[4] For training

[4] https://github.com/openai/gpt-2-output-dataset

ARGENT, we use the training and validation splits provided in WebText. For testing, we use the annotated WebText test set introduced by Pillutla et al. (2021) (Mauve paper), which includes synthetic texts generated by eight different language models. In this test set, human annotation is performed via pairwise comparisons of texts generated from different models on three criteria: human-like, sensible, and interesting. These pairwise judgments are aggregated into an overall ranking of generative models (model-wise ranking) by fitting a Bradley-Terry (BT) model (Marden, 1996).

We evaluate ARGENT across outputs from all eight generative models included in the Mauve test set. To enable direct comparison with results reported by Pillutla et al. (2021), we compute model-level scores by averaging ARGENT's predicted scores across all texts generated by each model. We then calculate the Spearman rank correlation between this machine-generated ranking and the human-derived ranking as used in the Mauve paper. Spearman correlation ranges from –1 to 1, with higher positive values indicating stronger alignment between the automatic and human rankings. It is important to interpret this metric with caution, as the correlation is computed over only eight ranked items, an insufficient sample size for drawing strong statistical conclusions.

Results Table 1 reports the Spearman correlations between ARGENT and human judgments, alongside six previously published evaluation models. We report results for the best-performing ARGENT variant, which was trained using local shuffling with a corruption probability range of 0–0.8 and a count-based scoring method (see Appendix C, Table 5, for results from other configurations). From the results, we can see that ARGENT achieved the second-highest performance for every criteria, just behind the Mauve model. However, Mauve has two key limitations when compared to ARGENT. First, it requires a human-generated corpus for evaluation whereas ARGENT only requires synthetic texts after it is pre-trained. Mauve directly measures distributional similarity between synthetic and human corpora, while ARGENT was trained in a zero-shot manner on corrupted real text that is different from the synthetic data used for testing. Second, it produces a single score per generative model, whereas ARGENT assigns a score to each individual output (we averaged ARGENT's per-text scores to obtain model-level scores for the purpose of comparison). Among the three evaluation criteria, Sensible is most closely aligned with language quality, where ARGENT performs comparably to Mauve.

4.2 Formal Text Evaluation: Synthetic Academic Publications

Data and Metrics We evaluate performance on formal text using the fully generated academic papers dataset from Liyanage et al. (2022), which contains 100 synthetic papers. We compare the performance of the same ARGENT trained on WebText data, with evaluation models reported in Liyanage et al. (2022), which includes BERT-based models trained on news headlines (Brown et al., 2020). Evaluating academic texts using an auto-evaluation model trained on informal WebText data allows us to assess ARGENT's generalisability across different domains.

Model	**Accuracy**
Bag of ngrams 1-3, MNBA (1)	19.7
Bag of ngrams 1-3, PACA (2)	31.8
Bag of ngrams 1-3, MCH (3)	19.7
Bag of ngrams 1-3, SVM (4)	39.7
LSTM model (Maronikolakis et al., 2021)	59.1
Bi-LSTM (Maronikolakis et al., 2021)	40.9
BERT (Maronikolakis et al., 2021)	52.5
DistillBERT (Maronikolakis et al., 2021)	62.5
ARGENT	**97.0**

Table 2: Performance of different evaluation models on academic publications. Liyanage et al. (2022) used Bag of ngrams as features for (1) MNBA - Multinomial Naive Bayes Algorithm (2) PACA - Passive Aggressive Classifier Algorithm (3) MCH - Multinomial Classifier with Hyperparameter (4) SVM - Support Vector Machine

Results The best performance was achieved by ARGENT using inflection-based corruption with a probability range of 0–0.6 and BLEURT-based scoring. Results for additional ARGENT configurations are provided in Appendix D Table 6. Table 2 presents these results alongside those of other evaluation models from the literature. Despite the domain mismatch, ARGENT shows the best performance among all models with a large margin, which demonstrates strong adaptability of ARGENT model.

4.3 Domain-specific Text Evaluation: Clinical Text

Data and Metrics To evaluate ARGENT's performance on domain-specific text, we generated

Metric	Gen. PPL	Zipf Coef.	REP	Distinct-4	Self-BLEU	Mauve	ARGENT
Human-like	81.0	83.3	-16.7	73.8	59.5	**95.2**	85.7
Sensible	73.8	69.0	-7.10	59.5	52.4	**85.7**	81.0
Interesting	64.3	52.4	-14.3	52.4	40.5	**81.0**	73.8

Table 1: Performance of different evaluation models on WebText (1) Generative perplexity (Fan et al., 2018) (2) Zipf Coefficient (Holtzman et al., 2020) (3) Repetition (Pillutla et al., 2021) (4) Distinct 4 n-grams (Pillutla et al., 2021) (5) Self-BLEU (Zhu et al., 2018) (6) Mauve (Pillutla et al., 2021)

synthetic clinical reports using BioGPT (Luo et al., 2022), which is fine-tuned on real clinical notes from a large secondary healthcare provider in the UK (Zecevic et al., 2024). Synthetic clinical text is an ideal use case, as access to real data in healthcare is often limited due to privacy and ethical constraints. In such contexts, synthetic clinical text can be valuable for NLP development, pre-training, and educational use. We generated a total of 97,152 clinical reports, using 92,652 for training and holding out 4,500 for testing. The dataset includes five types of clinical reports; details of these report types and the training/validation splits are provided in Appendix E Table 7. For evaluation, we computed the area under the pick-up rate curves, introduced in Section 3.2, across 10 batches for each report type. Each batch contained 100 reports, 90 synthetic and 10 real. We report the overall performance averaged across all report types here. Detailed results for each report type are provided in Appendix E.

Results The results of the grid search over corruption probability ranges for each evaluation method are provided in Appendix E, Table 8. The best-performing probability ranges for each configuration are as follows: inflection with count-based scoring: 0-0.4; inflection with BLEURT scoring: 0-1.0; shuffling count based: 0-0.4; shuffling BLEURT-based: 0-1.0; shufflection count-based: shuffling 0-0.6 and inflection 0-1.0; shufflection BLEURT-based: shuffling 0-0.8 and inflection 0-1.0. Table 3 presents the best overall performance for each ARGENT variant. The top-performing model is the shuffling-based variant with count-based scoring, achieving a pick-up rate AUC of 79.3%, substantially above the 50% random baseline. These results demonstrate that ARGENT can be effectively applied to domain-specific clinical text evaluation.

ARGENT models	Score
Inflection_count	68.1±2.4
shuffling_count	**79.3±2.6**
shufflection_count	67.7±3.5
Inflection_bleurt	58.7±5.8
shuffling_bleurt	56.8±6.4
shufflection_bleurt	59.4±6.1

Table 3: Performance of different ARGENT auto-evaluation models on clinical reports

5 Literature Review

Previous surveys of evaluation research (Yuan et al., 2021; Zhou et al., 2023) have typically classified evaluation methods based on task types or metric methodologies. For example, Yuan et al. (2021) grouped methods into supervised, unsupervised, and automatic evaluation metrics, while Zhou et al. (2023) classified evaluation studies according to the types of input and output involved in the task.

In contrast, our review is structured around the two core dimensions of our evaluation framework: (1) how references are selected, and (2) how similarity scores are defined. This perspective allows us to bridge reflective and open-ended generation tasks, and to analyse existing methods through the lens of reference construction and similarity function design.

5.1 Gold-standard reference selection

In RGT evaluation, references typically fall into two categories: pre-written human references and output-oriented references.

Pre-written References: Most evaluation studies rely on pre-written human references, often using multiple references to mitigate the limitations of any single gold standard. Many shared-task datasets provide such references. For instance, the WMT dataset[5], a widely used bench-

[5] https://www.statmt.org/wmt22/metrics/index.html

mark for machine translation evaluation, supplies a set of reference translations for each task. These are used in studies such as BERTScore (Zhang et al., 2020), BLEURT (Sellam et al., 2020), and BartScore (Yuan et al., 2021). However, little research has been done to justify or critically examine the selection process for pre-written references.

Output-Oriented References: Some studies adopt output-oriented references, also referred to as human-in-the-loop or human-targeted references (Snover et al., 2006). In this approach, human annotators manually edit model outputs to make them fluent and semantically equivalent to the intended input. These corrected outputs then serve as references for evaluation. For example, Snover et al. (2006) compare similarity scores between human-targeted and pre-written references using BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and TER (Przybocki et al., 2006), and show that human-targeted references yield higher correlations with human judgments across all three metrics.

This aligns with the discussions in this paper, which emphasises the importance of reference selection in determining evaluation quality. However, to our knowledge, the application of output-oriented reference construction to OGTs has not been explored in the literature.

5.2 Similarity Metrics

There is a substantial body of research on similarity metrics, which can broadly be divided into two categories: supervised methods, trained on human judgment as a regression task, and unsupervised methods, based on surface-level or semantic overlap between generated texts and references. These metrics may rely on either statistical features or neural embeddings.

Unsupervised Metrics: Statistical feature-based metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) measure similarity by counting overlapping *n*-grams between the output and reference. TER (Przybocki et al., 2006) uses edit distance to quantify dissimilarity. Embedding-based unsupervised metrics leverage neural encoders to project texts into vector space and compare their representations. For instance, BERTScore (Zhang et al., 2020) uses a BERT model to generate contextual embeddings for each token, and computes precision, recall, and F1 scores of the generative model based on the cosine similarity between the model outputs and reference embeddings. MoverScore (Zhao et al., 2019) extends this idea by computing the Earth Mover's Distance between the sets of token embeddings in the output and reference. This allows for soft alignment between tokens and better captures semantic similarity, especially in cases of paraphrasing or lexical variation.

Supervised Metrics: Supervised evaluation metrics are trained to predict human judgment. Stanojević and Sima'an (2014) propose BEER, a linear model that combines hand-crafted statistical features and is tuned using human annotations. BLEURT (Sellam et al., 2020) fine-tunes a BERT model to predict human evaluation scores based on the embeddings of output and reference sequences. COMET (Rei et al., 2020a) uses the XLM-RoBERTa (Conneau and Lample, 2019) encoder with pooling layers, fine-tuned on human preference rankings. These models generally achieve higher correlation with human judgment, but are limited by the training data domain and annotation quality.

5.3 Other evaluations

Proxy metrics Proxy metrics evaluate specific aspects of generated text that serve as indirect indicators of quality. For example, entity and relation coverage (Goodrich et al., 2019) or text length and token distribution (Yue et al., 2023) can be used to assess how well generated texts align with expected patterns. However, these metrics focus only on isolated properties of the output and do not provide a holistic measure of the generated texts.

Corpus Level metrics Corpus-level evaluation is widely adopted in OGT. These metrics compare the distribution of model-generated texts to that of human-written corpora using statistical properties. Examples include diversity of *n*-grams (e.g., Self-BLEU (Zhu et al., 2018)), generation perplexity (Fan et al., 2018) and repetition frequency (Holtzman et al., 2020), which measures how well the generated texts align with human language patterns. Mauve (Pillutla et al., 2021) introduces a KL-divergence-based metric to measure the divergence between distributions of model and human texts. However, these methods operate at the corpus level and do not provide scores for each document.

This work To the best of our knowledge, ARGENT is unique among existing evaluation methods. Unlike reference-based metrics, which require access to gold-standard texts, and unlike QE models, which rely on both the input (e.g., source text or prompt) and the output to predict quality, ARGENT operates solely on the output text. Rather than identifying a reference for a given text, we pre-train a model on a dataset composed of proxy model outputs paired with their most similar references and associated similarity scores. The model learns to map the proxy outputs directly to similarity scores without accessing the underlying references. During inference, ARGENT applies this learned ability to outputs from unseen text generation models, assigning a score that reflects the quality of the generated text.

6 Conclusion

In this work, we proposed a unified framework for evaluating machine-generated text that applies to both RGTs and OGTs. Building on this framework, we developed ARGENT, a novel reference-free auto-evaluation method for assessing the language quality of open-ended generation. ARGENT requires no human annotation and operates without relying on source inputs or reference corpora. We evaluated ARGENT across diverse text types and benchmarked it against several commonly used evaluation methods. Our results show that ARGENT outperforms all competing models except for Mauve on the WebText dataset, where it ranks second. However, unlike Mauve, ARGENT does not require a human reference corpus during evaluation and can assign quality scores at the level of individual outputs, rather than only at the model level. Finally, we reviewed the existing evaluation literature through the lens of our proposed framework, categorising prior methods based on reference selection strategies and similarity metric design.

7 Limitations

This paper introduces a text corruption pre-training method as a proxy for synthetic text, but only explores inflection and local shuffling as corruption methods. Targeted corruption strategies, designed to simulate specific evaluation criteria or mimic common errors found in synthetic text, could further improve the performance of auto-evaluation models.

Our experiments focus exclusively on evaluating the linguistic quality of generated texts. While language errors are common in earlier models, more advanced generative systems tend to exhibit issues such as overly generic or machine-like responses, as well as hallucinations. Extending the corruption-based training approach to address these types of errors presents an important avenue for future work.

8 Ethical Considerations

Although this work focuses on evaluating generated text rather than generating it, the implications of introducing a new evaluation metric like ARGENT can be important in measuring the performance of and ultimately optimising text generation models.

- ARGENT provides a scalable, reference-free method for estimating language quality in generated texts. Its accessibility and simplicity may encourage adoption for generation tasks.
- However, ARGENT is designed specifically to assess surface-level language quality, and does not evaluate other critical dimensions such as factual accuracy, harmful content, or social bias. Users should not over-interpret ARGENT scores as comprehensive measures of output quality and should use it in combination with other task-specific evaluations.

Use of the GSTT dataset received ethical approval from GSTT Electronic Records Research Interface (GERRI) institutional board review (IRAS ID = 257283). The reports were stored and processed in an approved, secure environment by authorised researchers. We do not report any individual data from the reports.

9 Acknowledgements

The research described in this paper was funded by King's College London DRIVE-Health Centre for Doctoral Training. We would like to express our gratitude to King's College London Computational Research Engineering and Technology Environment - Trusted Research Environment (CREATE-TRE) for providing compute resources and infrastructure.

References

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.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 9347–9359.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A Smith. 2021. All that's 'human'is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7282–7296.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

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.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898.

Ben Goodrich, Vinay Rao, Peter J Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 166–175.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. 2021. The perils of using mechanical turk to evaluate open-ended text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1265–1285.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Vijini Liyanage, Davide Buscaldi, and Adeline Nazarenko. 2022. A benchmark corpus for the detection of automatically generated text in academic publications. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4692–4700.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in bioinformatics*, 23(6):bbac409.

John I Marden. 1996. *Analyzing and modeling rank data*. CRC Press.

Antonis Maronikolakis, Hinrich Schütze, and Mark Stevenson. 2021. Identifying automatically generated headlines using transformers. In *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 1–6.

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 of the Association for Computational Linguistics*, pages 311–318.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Mark A Przybocki, Gregory A Sanders, and Audrey N Le. 2006. Edit distance: A metric for machine translation evaluation. In *LREC*, pages 2038–2043.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020a. Comet: A neural framework for mt evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020b. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 911–920, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231.

Miloš Stanojević and Khalil Sima'an. 2014. Beer: Better evaluation as ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 414–419.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Xiang Yue, Huseyin Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. 2023. Synthetic text generation with differential privacy: A simple and practical recipe. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1321–1342.

Agathe Zecevic, Xinyue Zhang, Sebastian Zeki, and Angus Roberts. 2024. Generation and evaluation of synthetic endoscopy free-text reports with differential privacy. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 14–24.

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*.

Haofei Zhao, Yilun Liu, Shimin Tao, Weibin Meng, Yimeng Chen, Xiang Geng, Chang Su, Min Zhang, and Hao Yang. 2024. From handcrafted features to llms: A brief survey for machine translation quality estimation. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE.

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.

Yongxin Zhou, Fabien Ringeval, and François Portet. 2023. A survey of evaluation methods of generated medical textual reports. In *Proceedings of the 5th Clinical Natural Language Processing Workshop*, pages 447–459.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

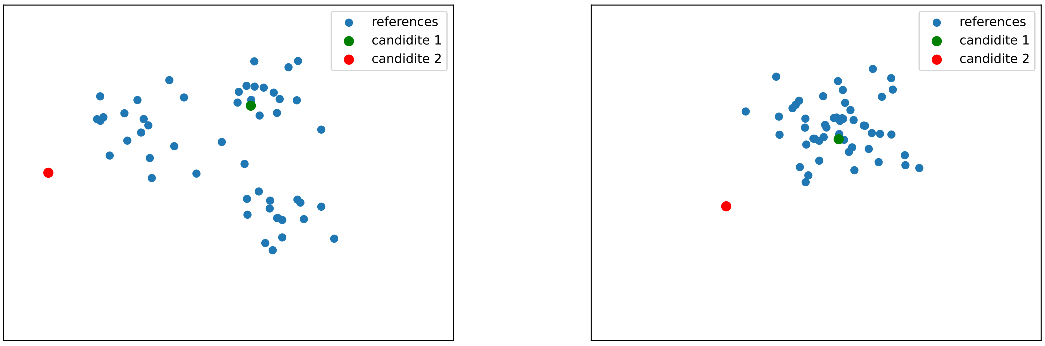
A Effects of references and similarity functions

To illustrate the importance of reference choice in evaluating generative tasks, we consider the following simple task, translation of the French sentence "C'est vraiment un homme intelligent" into English. Let us assume that we are comparing two models. Model 1 output is "He truly a smart man". This is largely correct, but missing the verb. Model 2 output is "He truly is a clever dog", with the noun completely wrong. Table 4 lists a set of possible correct translations (references) and the scores from different metrics comparing the outputs against these references. From the table, we can see: 1) Evaluation metrics can vary significantly based on the references used. If the last reference is used for evaluation, then with all three metrics, "He truly is a clever dog" will be picked as a better answer. 2) With BERTScore, the differences between references are smaller than with BLEU and ROUGE. This demonstrates that better metrics, such as those that take in to account semantics, can reduce variability caused by different references and thus may alleviate the problems caused by these.

References	BLEU	ROUGE-L	BERTScore
Candidate 1:**He truly a smart man**			
He truly is a smart man	82.24	90.91	96.14
He really is a smart guy	45.42	54.55	93.62
He really is an intelligent guy	18.18	0.50	93.30
He truly is a clever man	49.45	72.73	94.98
Candidate 2: **He truly is a clever dog**			
He truly is a smart man	55.68	66.67	94.72
He really is a smart guy	37.95	50.00	92.98
He really is an intelligent guy	26.04	33.33	92.62
He truly is a clever man	82.94	83.33	95.45

Table 4: Scores of two translation candidates against different references with different metrics

The illustrative graph 3 visualises the effects of references and similarity functions. The graph shows a toy 2-D version of space where the Euclidean distance between two points in this graph represents the similarity score between the points defined by some similarity function. In each space, blue dots represent all the gold-standard references, with two candidates of machine output are marked by green and red. In this graph, we can see that the red point is a worse candidate compare to red. But if we chose the left most reference, then the red point would have a higher score. For example, this could be the case in our example where the "He truly is a clever dog" translation scores higher with certain references. But according to our evaluation theory, the score of the green candidate should be defined by the blue dot closest to it which is the one right on top of it, and the score of the red candidate is defined by the closest blue dot on its right. This will give us a correct judgement that the green candidate is a better candidate than the red one. 3(b) shows a space using a better similarity function for example, BERT score versus BLEU. we can see that this similarity function has better ability to cluster the acceptable references closer than 3(a), This reduces the variability in the scores due to different reference choices. In this graph, if we chose the reference on the left, the distance to the red dot is not so close compared to that to the green one. But this may not solve the problem. The selection of the closest reference is still not replaceable in most tasks, especially those with large reference spaces.



(a) some similarity function space

(b) a better similarity function space

Figure 3: Illustration of effects of reference points and similarity function

B Text Corruption Methods

Algorithm 1 Token Inflection

```
Define pos_list, inflection_probability, initialise inflected_text ← empty string ""
for current_token in text do
    if draw from inflection_probability then
        current_pos ← pos_tagger(sentence, current_token)
        inflected_pos ← pos_list - current_pos
        inflected_token ← inflection(token, inflected_pos)
        inflected_text ← inflected_text+" "+inflected_token
    end if
end for
return inflected_text
```

Algorithm 2 Token shuffling

```
Define window_range, shuffling_probability, initialise shuffled_text ← empty string "", remain_text ← text
while len(remain_text)>0 do
    if draw from shuffling_probability then
        draw win_length from window_range
        curr_text←remain_text[:win_length]
        shuffled_text ← shuffled_text +" "+ shuffle(current_text)
        remain_text ← remain_text-curr_text
    end if
end while
return shuffled_text
```

Algorithm 3 Text Corruption with corruption count based score

```
Define corruption method set K, prob range $p_{range}$, initialise corr_data
for text n in N do
    initialise corr_count = 0
    for corruption method k in K do
        prob ← random(0, prob_range)
        corr_text = corr_method_k(text, prob)
        for i in text length do
            if corr_text[i] != text[i] then
                corr_count ← corr_count + 1
            end if
        end for
    end for
    score = 1-corr_count/len(K)*N
    corr_data append (corr_text, score)
end for
return corr_data
```

C Hyper-parameter tuning for WebText evaluation

Score	Prob	Inflection			Shuffling		
		Human-like	Sensible	Interesting	Human-like	Sensible	Interesting
Count	0-0.2	83.3	71.4	69.0	0-0.2	85.7	81.0
	0-0.4	83.3	71.4	69.0	78.6	76.2	61.9
	0-0.6	69.0	57.1	45.2	81.0	73.8	66.7
	0-0.8	83.3	76.2	69.0	85.7	81.0	73.8
	0-1.0	66.7	52.4	54.8	81.0	78.6	66.7
BLEURT	0-0.2	-47.6	-52.4	-61.9	-40.0	-45.0	-51.7
	0-0.4	47.6	35.7	35.7	-59.5	-64.3	-81.0
	0-0.6	64.3	54.8	52.4	-9.52	-14.3	-40.5
	0-0.8	81.0	73.8	66.7	-90.5	-90.5	-97.6
	0-1.0	81.0	73.8	66.7	-38.1	-40.0	-57.1
Shufflection (Prob: Shuffling, Inflection)							
Count	0-0.2, 0-0.4	88.1	78.6	76.2	86.7	80.0 3	76.7
	0-0.2, 0-0.8	88.1	78.6	76.2	70	61.7	60
	0-0.8, 0-0.4	88.1	78.6	76.2	79.9	71.7	66.7
	0-0.8, 0-0.8	85.7	76.2	71.4	78.36	70.0	63.3

Table 5: Hyper-parameter tuning: inflection on webtext data

Table 5 shows no great differences between shuffling and inflection. Interestingly, a BLEURT-based score does not give a high score in most cases

D Hyper-parameter Tuning for Synthetic Academic Publications

method	score	0-0.2	0-0.4	0-0.6	0-0.8	0-1.0
Inflection	Count	58	52	59	51	52
	BLEURT	85	79	97	86	80
Shuffling	Count	69	69	68	67	63
	BLEURT	93	77	64	91	75

Table 6: Hyper-parameter tuning: synthetic academic publications

From the Table 6, we can see that the model using BLEURT-based score tends to be the best for this task, and the difference of using inflection or shuffling method is not very significant.

E Hyper-parameter tuning for clinical text evaluation

The clinical reports include five types: Colonoscopy, Gastroscopy, Endoscopic ultrasound (EUS), Sigmodoiscopy and Endoscopic Retrograde Cholangiopancreatography (ERCP). The number of training and testing samples for each type can be found in Table 7. Table 8 shows that with count-based score models, the performance for colonoscopy, gastroscopy and flexible sigmoidoscopy tends to be better than the performance of EUS and ERPC.

Model	**Prob**	**Col**	**Endo**	**ERCP**	**Gstr**	**Sig**	**Total**
train	20411	2009	1348	40658	9453	243	74122
valid	3676	971	784	10263	2790	46	18530
total	24087	2980	2132	50948	12243	289	92652

Table 7: Statistics of clinical data

Score	**Prob**	**Col**	**Endo**	**ERCP**	**Gstr**	**Sig**	**Total**
Inflection							
Count	0-0.2	66.1±7.9	60.5±10.6	58.0±9.9	67.9±11.2	67.5±13.8	64.0±4.7
	0-0.4	70.1±6.6	62.9±10.5	64.6±12.7	70.9±9.3	71.8±10.9	**68.1±2.4**
	0-0.6	66.9±6.1	56.0±11.3	61.8±10.4	66.9±11.0	72.1±10.6	64.7±4.2
	0-0.8	68.8±8.8	62.4±11.1	61.7±10.1	70.6±8.3	71.0±9.3	66.9±2.9
	0-1.0	69.6±5.6	59.6±13.0	62.9±9.3	72.6±10.2	70.7±9.0	67.1±3.1
BLEURT	0-0.2	58.1±12.1	56.1±9.8	56.2±9.2	61.3±15.6	54.8±11.0	57.3±6.3
	0-0.4	59.1±12.3	55.5±10.0	54.2±10.0	60.1±16.0	54.8±11.0	56.7±6.1
	0-0.6	59.3±12.3	54.8±9.2	54.5±9.3	60.4±15.0	57.0±11.4	57.2±5.8
	0-0.8	60.4±12.3	56.5±10.2	56.1±8.9	60.4±15.3	56.7±10.9	58.0±6.4
	0-1.0	60.5±11.1	56.4±9.4	58.5±9.2	60.9±14.9	57.0±10.4	**58.7±5.8**
Shuffling							
Count	0-0.2	66.1±8.5	63.7±11.3	62.2±10.7	69.7±13.9	67.7±12.9	65.9±3.8
	0-0.4	82.9±8.2	76.3±8.0	74.0±7.6	81.6±9.8	81.7±12.0	**79.3±2.6**
	0-0.6	74.6±5.7	60.9±10.7	67.4±8.4	73.9±12.1	73.5±10.2	70.0±2.6
	0-0.8	64.9±7.8	58.4±8.5	61.2±10.1	65.4±13.8	60.5±12.5	62.1±2.6
	0-1.0	71.6±8.4	66.7±10.6	67.9±10.2	75.1±13.0	68.4±13.5	69.9±3.4
BLEURT	0-0.2	54.8±14.5	55.4±9.5	58.7±8.1	59.0±15.6	53.1±10.4	56.2±6.2
	0-0.6	54.2±14.1	55.7±9.4	58.8±8.6	58.6±15.6	53.9±10.5	56.2±6.2
	0-0.6	54.5±14.5	55.8±10.6	59.7±6.7	58.2±15.5	53.6±10.2	56.3±6.4
	0-0.8	55.7±13.1	54.8±10.2	59.2±8.1	59.5±16.1	53.7±9.6	56.6±6.0
	0-1.0	54.4±13.7	55.3±10.4	59.8±8.3	59.6±15.1	55.0±10.0	**56.8±6.4**
Shufflection (Prob: Shuffling, Inflection)							
Count	0-0.4, 0-0.4	64.6±7.4	60.2±7.4	62.1±10.0	67.1±15.4	64.8±11.4	63.8±3.2
	0-0.4, 0-1.0	66.6±7.6	57.4±8.3	62.1±11.1	68.2±12.6	63.4±11.4	63.9±3.1
	0-0.6, 0-0.4	66.3±6.8	59.8±9.0	60.9±9.3	66.6±13.4	64.6±10.4	63.6±3.3
	0-0.6, 0-1.0	80.6±8.1	57.2±6.2	64.3±11.1	69.1±13.6	67.3±11.7	**67.7±3.5**
BLEURT	0-1.0, 0-1.0	58.3±11.8	56.4±10.5	59.5±74.1	59.6±16.2	57.4±10.5	58.2±6.4
	0-1.0, 0-0.8	60.4±13.5	55.8±11.7	59.7±8.5	62.1±15.3	58.6±9.7	59.3±6.3
	0-0.8, 0-1.0	60.5±12.2	57.1±9.9	59.2±9.0	62.0±14.2	58.1±9.9	**59.4±6.1**
	0-0.8, 0-0.8	60.7±11.9	55.4±9.7	59.3±8.7	61.0±16.2	57.5±9.9	58.8±5.6

Table 8: Hyper-parameter tuning on clinical reports

F License Use Information

We confirm that all external datasets and software tools used in this work comply with their respective licenses and have been used in accordance with intended purposes:

- The Mauve-annotated dataset (Pillutla et al., 2021) and the synthetic academic paper dataset (Liyanage et al., 2022) are used under the GNU General Public License v2.0.
- BLEU (Papineni et al., 2002) is used under the BSD 3-Clause License.
- ROUGE (Lin, 2004) and BLEURT (Sellam et al., 2020) are used under the Apache License 2.0.
- BERTScore (Zhang et al., 2020) is used under the MIT License.