Active Evaluation: Efficient NLG Evaluation with Few Pairwise Comparisons

Anonymous ACL submission

Abstract

Recent studies have shown the advantages of evaluating NLG systems using pairwise comparisons as opposed to direct assessment. Given k systems, a naive approach for identifying the top-ranked system would be to uniformly obtain pairwise comparisons from all $\binom{k}{2}$ pairs of systems. However, this can be very expensive as the number of human annotations required would grow quadratically with k. In this work, we introduce *Active Evaluation*, a framework to efficiently identify the top-ranked system by actively choosing system pairs for comparison using dueling bandit algorithms. We perform extensive experiments with 13 dueling bandits algorithms on 13 NLG evaluation datasets spanning 5 tasks and show that the number of human annotations can be reduced by 80%. To further reduce the number of human annotations, we propose model-based dueling bandit algorithms which combine automatic evaluation metrics with human evaluations. Specifically, we eliminate sub-optimal systems even before the human annotation process and perform human evaluations only on test examples where the automatic metric is highly uncertain. This reduces the number of human annotations required *further* by 89%. In effect, we show that identifying the top-ranked system requires only a few hundred human annotations, which grow linearly with k. Lastly, we provide practical recommendations and best practices to identify the top-ranked system efficiently. [1]

1 Introduction

In the last few years, the field of NLG has made rapid progress with the advent of large-scale models trained on massive amounts of data (Vaswani et al., 2017; Xue et al., 2020; Liu et al., 2020; Brown et al., 2020). However, evaluation of NLG systems continues to be a challenge. On the one hand, we have automatic evaluation metrics which are easy to compute but unreliable. In particular, many studies have shown that they do not correlate well with human judgments (Novikova et al., 2017; Elliott and Keller, 2014; Sai et al., 2019, 2020a,b). On the other hand, we have human evaluations, which are relatively more reliable but tedious, expensive, and time-consuming. Further, recent studies have highlighted some limitations of human evaluations that involve direct assessment on an absolute scale, *e.g.*, Likert scale. Specifically, human evaluations using direct assessment have been shown to suffer from *annotator bias*, *high variance* and *sequence effects* where the annotation of one item is influenced by preceding items (Kulikov et al., 2019; Sudoh et al., 2021; Liang et al., 2020; See et al., 2019; Mathur et al., 2017).

In this work, we focus on reducing the cost and time required for human evaluations while not compromising on reliability. We take motivation from studies which show that selecting the better of two options is much easier for human annotators than providing an absolute score, which requires annotators to maintain a consistent standard across samples (Kendall, 1948; Simpson and Gurevych, 2018). In particular, recent works show that ranking NLG systems using pairwise comparisons is a more reliable alternative than using direct assessment (See et al., 2019; Li et al., 2019; Sedoc et al., 2019; Dhingra et al., 2019). While this is promising, a naive approach for identifying the top-ranked system from a set of k systems using uniform exploration is prohibitively expensive. Specifically, uniform exploration obtains an equal number of annotations for all the $\binom{k}{2}$ system pairs; as a result, the required human annotations grows as $O(k^2)$.

To reduce the number of pairwise annotations, we introduce Active Evaluation, a framework to efficiently identify the top-ranked NLG system. Our Active Evaluation framework consists of a learner that selects a pair of systems to compare

[1] Our code and trained model checkpoints will be made publicly available

at each time step. The learner, then, receives a feedback signal indicating the (human) preference between the selected systems on one input context, randomly sampled from the test dataset. The learner's objective is to reliably compute the top-ranked system with as few human annotations as possible. We adopt algorithms from the stochastic dueling bandits literature (Bengs et al., 2021) to decide which pair of NLG systems to compare at each time step. To check if existing dueling bandits algorithms can indeed provide reliable top-rank estimates with minimal annotations, we evaluate 13 such algorithms on 13 NLG evaluation datasets spanning five tasks *viz.*, machine translation, summarization, data-to-text generation, paraphrase generation, and grammatical error correction. We show that the best performing dueling bandit algorithm can reduce the number of human annotations by 80% when compared to uniform exploration.

To further reduce human annotations, we leverage automatic evaluation metrics in our Active Evaluation framework. We utilize existing automatic metrics such as BLEU (Papineni et al., 2002), BertScore (Zhang et al., 2020), *etc* for pairwise evaluations by converting the direct evaluation scores into preference probabilities using pairwise probability models. We also develop trained pairwise metrics that directly predict the comparison outcome given pairs of generated texts and context or reference as input. To incorporate such evaluation metrics in our Active Evaluation framework, we propose three model-based dueling bandits algorithms, *viz.*, (i) Random Mixing: human annotations and evaluation metric predictions are randomly mixed, (ii) Uncertainty-aware selection: human annotations are obtained only when the predictions from the evaluation metric is highly uncertain, (iii) UCB Elimination: poorly performing NLG systems are eliminated using an Upper Confidence Bound (UCB) on the evaluation metric scores. Through our experiments, we show that the number of human annotations can be further reduced by 89% on average (this reduction is over and above the 80% reduction that we got earlier). In effect, we show that given k systems, we can find the top-ranked NLG system efficiently with just a few hundred comparisons that vary as $O(k)$. Lastly, we provide practical recommendations to efficiently identify the top-ranked NLG system based on our empirical study on various design choices and hyperparameters.

2 Active Evaluation Framework

We introduce the problem and our Active Evaluation setup in section 2.1. Later in section 2.2, we describe the different approaches to decide which pairs of NLG systems to compare at each time step. Finally, in section 2.3, we formalize the notion of top-ranked system.

2.1 Problem Formulation and Setup

We consider the problem of finding the top-ranked NLG system from a given set of k systems, denoted by $\mathcal{S} = \{1, 2, \ldots, k\}$. Our Active Evaluation framework consist of a *leaner* which at each time step t, chooses a pair of systems $s_1^{(t)}, s_2^{(t)} \in \mathcal{S}$ for comparison. Then, we ask human annotators to compare the outputs of the chosen systems on a randomly sampled input context and provide the comparison outcome as feedback to the learner. Specifically, we first sample an input context $X^{(t)}$ from the test dataset and obtain the generated texts $Y_1^{(t)}, Y_2^{(t)}$ from the chosen systems $s_1^{(t)}, s_2^{(t)}$. We then display the generated texts $Y_1^{(t)}, Y_2^{(t)}$ along with the context $X^{(t)}$ to human annotators and obtain a comparison outcome $w^{(t)} = 1, 0$, or 0.5 denoting whether $Y_1^{(t)}$ is of better, worse, or equal (tie) quality as $Y_2^{(t)}$. Note that the feedback $w^{(t)}$ indicates the preference on only one input sample and not the entire test dataset. The overall framework is depicted in figure 1. The learner's objective is to find the top-ranked system with as few pairwise comparisons as possible.

2.2 Choosing System Pairs for Comparison

The learner should decide the pair of systems $(s_1^{(t)}, s_2^{(t)})$ to compare at each time step t. The naive approach is to uniformly explore all the $\binom{k}{2}$ system pairs. Specifically, the probability of selecting a pair $(i, j), i \neq j$ at time t is given by

$$P_{uniform}((s_1^{(t)}, s_2^{(t)}) = (i, j)) = \frac{1}{\binom{k}{2}}$$

However, as we show in our experiments, the number of human annotations required to find the top-ranked system by this approach is very expensive and grows quadratically with the number of systems since we equally explore all $\binom{k}{2}$ pairs. To reduce the number of annotations, we use dueling bandit algorithms to actively choose pairs of systems to compare based on the history of previous observations. We provide an overview of 13 dueling bandits algorithms proposed in the literature in

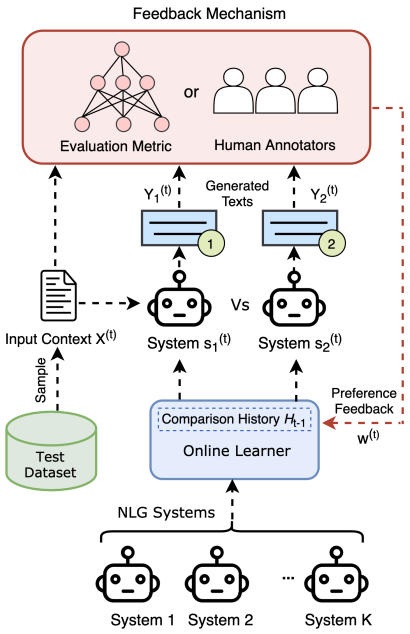

Figure 1: Our Active Evaluation framework consisting of a learner that chooses a pair of systems to compare at each time step. The learner receives feedback from either human annotators or the automatic metric.

appendix B. We refer the readers to (Bengs et al., 2021) for a complete survey.

2.3 Identifying the top-ranked system

We now formalize the notion of the top-ranked system. Let p_{ij} denote the preference probability of system i over system j *i.e.* the probability that a generated text from system i is preferred over system j in the test dataset. We say that a system i "beats" system j if $p_{ij} > \frac{1}{2}$. In other words, system i beats system j if the probability of winning in a pairwise comparison is larger for i than it is for j. We define the top-ranked system i^* as the one that beats all other systems, *i.e.* $p_{i^*j} > \frac{1}{2}, \forall j \in \mathcal{S} - i^*$.

3 Pairwise Probability Models

Our Active Evaluation framework, which we described in the previous section, completely relied on human annotators to compare pairs of generated texts (Y_1, Y_2) to provide the preference feedback w. We can further reduce the number of required human annotations by estimating the human preference feedback using automatic evaluation metrics. However, most existing evaluation metrics are designed for direct assessment and not directly suitable for pairwise evaluations. In this section, we describe three pairwise probability models to convert direct evaluation scores into pairwise preference probabilities. Let $f(Y)$ denote the score provided by a direct assessment metric f to a generated text Y (The dependence of f on the reference/context is omitted for brevity). The pairwise preference probability $\hat{p}(Y_1 \succ Y_2)$ between any two hypotheses Y_1 and Y_2 can be modeled in 3 different ways:

- **Linear:**

$$\hat{p}(Y_1 \succ Y_2) = \frac{1}{2} + (f(Y_1) - f(Y_2))$$

- **Bradley-Terry-Luce (BTL)** (Bradley and Terry, 1952; Luce, 1979):

$$\hat{p}(Y_1 \succ Y_2) = \frac{f(Y_1)}{f(Y_1) + f(Y_2)}$$

- **BTL-logistic:**:

$$\hat{p}(Y_1 \succ Y_2) = \frac{1}{1 + e^{(f(Y_1) - f(Y_2))}}$$

As detailed in appendix C.2, we appropriately preprocess the scores $f(Y)$ to ensure that preference probability lies between 0 and 1. We can now predict the comparison outcome w by thresholding the preference probability at two thresholds τ_1 and $\tau_2 (\geq \tau_1)$ to incorporate ties *i.e.*:

$$\hat{w} = \begin{cases} 1, & \text{if } \hat{p}(Y_1 \succ Y_2) > \tau_2 \\ 0, & \text{if } \hat{p}(Y_1 \succ Y_2) < \tau_1 \\ 0.5, & \text{Otherwise} \end{cases}$$

We choose τ_1 and τ_2 using grid search on the validation set. Refer appendix C.2 for more details.

4 Model-based Dueling Bandits

In the previous section, we discussed pairwise probability models to obtain the estimated preference probability $\hat{p}(Y_1 \succ Y_2)$ and the comparison outcome $\hat{w}$ using scores assigned by direct assessment metrics. We now propose three model-based dueling bandit algorithms wherein we combine such predictions from evaluation metrics with human annotations in the Active Evaluation framework.

4.1 Random Mixing

Here, we randomly provide either the real (human) or the evaluation metric predicted feedback to the learner. Specifically, at any time t, we use the predicted comparison outcome $\hat{w}^{(t)}$ as the feedback with probability p_m and use human annotations $w^{(t)}$ as feedback with probability $1 - p_m$. The hyperparameter p_m controls the ratio of estimated and real feedback given to the learner. As with other hyperparameters, we tune p_m on the validation set.

4.2 Uncertainty-aware Selection

In this algorithm, we estimate uncertainty in the evaluation metric predictions and decide to ask for human annotations only when the evaluation metric is highly uncertain. We specifically focus on trainable neural evaluation metrics such as Bleurt (Sellam et al., 2020) where we estimate the prediction uncertainty using recent advances in Bayesian deep learning. Let $\hat{p}(Y_1 \succ Y_2|\theta)$ denote the preference probability modelled by a neural evaluation metric with parameters θ. Given a training dataset $\mathcal{D}^{tr}$, Bayesian inference involves computing the posterior distribution $p(\theta|\mathcal{D}^{tr})$ and marginalization over the parameters θ:

$$\hat{p}(Y_1 \succ Y_2|\mathcal{D}^{tr}) = \int_{\theta} \hat{p}(Y_1 \succ Y_2|\theta)\hat{p}(\theta|\mathcal{D}^{tr})d\theta$$

However, computing the true posterior and averaging over all possible parameters is intractable in practice. Hence, several approximations have been proposed in variational inference such as finding a surrogate distribution $q_\phi(\theta)$ for the true posterior. Gal and Ghahramani (2016) have shown that we can use the Dropout distribution (Srivastava et al., 2014) as the approximate posterior $q_\phi(\theta)$. Specifically, we can perform approximate Bayesian inference by applying Dropout during test time. Hence, the posterior can now be approximated with Montecarlo samples as follows:

$$\hat{p}(Y_1 \succ Y_2|\mathcal{D}^{tr}) \approx \frac{1}{L}\sum_{l=1}^{L} \hat{p}(Y_1 \succ Y_2|\theta_l)$$

where $\{\theta_l\}_{l=1}^{L}$ are L samples from the Dropout distribution $q_\phi(\theta)$ (i.e. we apply Dropout L times independently during testing). We now discuss two different Bayesian uncertainty measures:

BALD: The Bayesian Active Learning by Disagreement (BALD) (Houlsby et al., 2011) is defined as the mutual information between the model predictions and the model posterior. Let $p_l = \hat{p}(Y_1 \succ Y_2|\theta_l)$, where $\theta_l \sim q_\phi(\theta)$, be the evaluation metric prediction using the l^{th} sample θ_l from the Dropout distribution. Also, let $\bar{p} = \frac{1}{L}\sum_{l=1}^{L} p_l$ be the mean prediction. As shown in (Gal et al., 2017), we can approximate the BALD measure using samples from the Dropout distribution as:

$$\hat{\mathbb{I}} = \mathbb{H}(\bar{p}) - \frac{1}{L}\sum_{l=1}^{L} \mathbb{H}(p_l)$$

where $\mathbb{H}$ is the binary cross entropy function. The BALD uncertainty score is essentially the difference in entropy of the mean prediction $\bar{p}$ and the average entropy of the individual predictions $\{p_l\}_{l=1}^{L}$. Hence, the BALD uncertainty score is high when the metric's mean prediction is uncertain (high entropy) but the individual predictions are highly confident (low entropy), *i.e.*, when the metric produces disagreeing predictions with high confidence.

STD: We also adopt the standard deviation of the preference probability taken over the posterior distribution as a measure of uncertainty:

$$\sigma = \sqrt{\mathrm{Var}_{\theta \sim \hat{p}(\theta|\mathcal{D}^{tr})}(\hat{p}(Y_1 \succ Y_2|\theta))}$$

Similar to BALD, we can approximate the above measure using the empirical standard deviation of samples drawn from the dropout distribution.

Our proposed algorithm asks for human annotations only if the uncertainty measure (BALD or STD) is above a particular threshold.

4.3 UCB Elimination

The key idea here is to eliminate a set of "poorly performing" NLG systems using the automatic metric and perform human evaluations with the remaining set of systems. To eliminate sub-optimal systems, we first need to quantify a performance measure for the systems. We use the Copeland score (Zoghi et al., 2015) which is defined as the normalized total number of pairwise wins for a system: $C_i = \frac{1}{k-1}\sum_{j \neq i} \mathbb{1}(p_{ij} > \frac{1}{2})$. Copeland score is the highest for the top-ranked system with a value of 1 and it is less than 1 for all other systems. To estimate the Copeland score, we first predict the pairwise preference probability between any two systems i and j as follows:

$$\hat{p}_{ij} = \frac{1}{N}\sum_{Y_1, Y_2 \in \mathcal{D}_{ij}} \hat{p}(Y_1 \succ Y_2|\theta)$$

where $\mathcal{D}_{ij}$ is the test dataset consisting of generated texts from systems i and j, N is the total number of test examples, θ is the learned model parameters. We can now estimate the Copeland score $\hat{C}_i$ using the estimated preference $\hat{p}_{ij}$ and eliminate all systems with Copeland scores below a threshold. However, a major problem with this approach is that evaluation metrics are often inaccurate and we could wrongly eliminate the true top-ranked system without performing any human evaluations. For example, consider the example where i^* is the

top-ranked system with $p_{i^*j} > 0.51\ , \forall j \in \mathcal{S} - i$. If several of the predicted probabilities $\hat{p}_{i^*j}$ are less than 0.5, our top-ranked system i^* will receive a low estimated Copeland score and will be incorrectly eliminated. To overcome this problem, we define an Upper Confidence Bound (UCB) on the preference probability using uncertainty estimates that we described in 4.2. Specifically, the upper confidence bound $\hat{u}_{ij}$ is given by $\hat{u}_{ij} = \hat{p}_{ij} + \alpha\hat{\sigma}_{ij}$ where α is a hyperparameter that controls the size of the confidence region and $\hat{\sigma}^2_{ij}$ is the estimated variance given by:

$$\hat{\sigma}^2_{ij} = \frac{1}{N^2} \sum_{Y_1, Y_2 \in \mathcal{D}_{ij}} \text{Var}_{\theta \sim q_\phi(\theta)} \hat{p}(Y_1 \succ Y_2 | \theta)$$

where $q_\phi(\theta)$ is the Dropout distribution. Using the upper confidence estimates $\hat{u}_{ij}$, we now define the optimistic Copeland score for a system i as $\hat{C}^u_i = \frac{1}{K-1} \sum_{j \neq i} \mathbb{1}(\hat{u}_{ij} > \frac{1}{2})$. Here, we consider a system i to beat another system j ($\hat{u}_{ij} > 0.5$) if either the estimated preference is high ($\hat{p}_{ij}$ is high) or if there is an high uncertainty in the estimation ($\hat{\sigma}_{ij}$ is high). In UCB Elimination, we eliminate a system only if the optimistic Copeland score is below a threshold.

5 Experimental Setup

In this section, we describe the (i) NLG tasks and datasets used in our experiments, (ii) automatic evaluation metrics used in our model-based algorithms, and (iii) annotation complexity measure used for comparing dueling bandit algorithms.

5.1 Tasks & Datasets

We use a total of 13 datasets spanning 5 tasks in our experiments which are summarized in table 1.

Machine Translation (MT): We use 7 human evaluation datasets collected from the WMT news translation tasks (Bojar et al., 2015, 2016) *viz.* fin→eng, rus→eng, deu→eng language pairs in WMT 2015 and tur→eng, ron→eng, cze→eng, deu→eng language pairs in WMT 2016.

Grammatical Error Correction (GEC): We utilize two human evaluation datasets collected by (Napoles et al., 2019) where the source texts are from (i) student essays (FCE), and (ii) formal articles in Wikipedia (Wiki). We also use another GEC dataset collected by (Napoles et al., 2015a) from the CoNLL-2014 Shared Task (Ng et al., 2014).

Data-to-Text Generation: We use the human evaluation data from the E2E NLG Challenge (Dusek et al., 2020). The task here is to generate natural language utterance from dialogue acts.

Paraphrase Generation: We use human evaluations of model generated English paraphrases released with the ParaBank dataset (Hu et al., 2019).

Summarization: We make use of the human evaluations (Stiennon et al., 2020) of GPT3-like transformers on the TL;DR dataset (Völske et al., 2017). We provide further details including preprocessing steps and downloadable links in appendix A.1.

Task	Dataset	# Systems	# Human Annotations
Machine Translation	WMT15 fin→eng	14	31577
	WMT15 rus→eng	13	44539
	WMT15 deu→eng	13	40535
	WMT16 tur→eng	9	10188
	WMT16 ron→eng	7	15822
	WMT16 cze→eng	12	125788
	WMT16 deu→eng	10	20937
Grammatical Error Correction	Grammarly (FCE)	7	20328
	Grammarly (Wiki)	7	20832
	CoNLL-2014 Shared Task	13	16209
Data-to-Text	E2E NLG Challenge	16	17089
Paraphrase	ParaBank	28	151148
Summarization	TLDR OpenAI	11	4809

Table 1: Description of tasks and datasets with the number of NLG systems and pairwise human annotations

5.2 Automatic NLG Evaluation Metrics

We can predict the comparison outcome w using two approaches. First, we can use pairwise probability models with existing direct assessment metrics as discussed in section 3. Alternatively, we can train evaluation metrics to directly predict the comparison outcome given pairs of generated texts and context/reference as input. We discuss both these approaches below:

Direct Assessment Metrics: We experiment with a total of 10 direct assessment metrics *viz.* chrF (Popovic, 2015), BLEU-4 (Papineni et al., 2002), ROUGE-L (Lin, 2004), Embedding Average (Wieting et al., 2016), Vector Extrema (Forgues et al., 2014), Greedy Matching (Rus and Lintean, 2012), Laser (Artetxe and Schwenk, 2019), BertScore (Zhang et al., 2020), MoverScore (Zhao et al., 2019) and Bleurt (Sellam et al., 2020). We mention the implementation details in appendix A.2.

Pairwise Evaluation Metrics: We finetune the pretrained Electra-base transformer model (Clark et al., 2020) to directly predict the comparison outcome w. We curate task-specific human evaluation datasets consisting of tuples of the form (context/reference, hypothesis 1, hypothesis 2, label) for finetuning. Due to space constraints, we mention

Algorithm	WMT 2016				WMT 2015			Grammarly		CoNLL '14 Task	E2E NLG	Para-Bank	TL; DR
	tur-eng	ron-eng	cze-eng	deu-eng	fin-eng	rus-eng	deu-eng	FCE	Wiki				
Uniform	19479	24647	10262	3032	2837	12265	17795	8115	34443	61369	65739	825211	5893
SAVAGE	10289	18016	6639	2393	2675	12806	12115	5767	22959	39208	41493	255208	4733
DTS	10089	9214	8618	4654	4850	13317	16473	4355	11530	18199	19940	170467	1354
CCB	7017	11267	5389	2884	4092	11548	10905	4386	10020	21392	16960	87138	2518
Knockout	3415	7889	4723	3444	5104	5809	5956	3134	3777	**8055**	7708	17418	4953
RUCB	3125	5697	3329	1636	**1655**	4536	6222	2732	5617	19024	10924	41149	1647
RCS	2442	**3924**	3370	1537	2662	3867	5296	**1816**	**4606**	12678	7263	34709	1903
RMED	**2028**	5113	**1612**	**864**	1707	**1929**	**4047**	2093	5647	9364	**3753**	**24132**	**1162**

Table 2: Annotation complexity of the top 7 best performing dueling bandit algorithms along with the uniform exploration algorithm on 13 datasets spanning 5 NLG tasks

details on the datasets and finetuning in appendix A.3 and A.4. For the summarization task alone, we couldn't find any pairwise human judgment dataset sufficient for finetuning the Electra model.

5.3 Annotation Complexity Measure

To evaluate the performance of dueling bandit algorithms, we define *annotation complexity* as the minimum number of human annotations needed by an algorithm to identify the top-ranked NLG system with high confidence. Let i^* be the actual top-ranked system, and $\hat{i^*}(n)$ denote the estimated winner by the algorithm after obtaining n human annotations, then query complexity is defined as:

$$\min n' : \forall n \geq n', P(\hat{i^*}(n) = i^*) > 1 - \delta_{acc}$$

where δ_{acc} is the allowable failure probability *i.e.* the learner can make a mistake with at most δ_{acc} probability. To compute the annotation complexity, we run each dueling bandit algorithm with 200 different random seeds and find the minimum number of human annotations after which the algorithm correctly returns the top-ranked NLG system in at least 190/200 runs (we set $\delta_{acc} = 0.05$).

6 Results & Discussion

We discuss the performance of dueling bandits algorithms in 6.1, automatic metrics in 6.2 and our proposed model-based algorithms in 6.3. Lastly in 6.4, we analyze the variation of annotation complexity with the number of NLG system.

6.1 Analysis of Dueling Bandit Algorithms

We report the annotation complexity of the top 7 dueling bandit algorithms along with uniform exploration on 13 datasets in table 2. We observe that the annotation complexity of uniform exploration is consistently high across all 13 datasets. In particular, the required human annotations become prohibitively expensive when the number of NLG systems is high, *e.g.* E2E NLG (16 systems) and ParaBank (28 systems) datasets. On the other hand, dueling bandit algorithms such as RUCB (Zoghi et al., 2014b), RCS (Zoghi et al., 2014a), RMED (Komiyama et al., 2015) are able to effectively identify the top-ranked system with much fewer annotations. In particular, RMED performs the best with a reduction of 80.01% in human annotations compared to uniform exploration. We also examine an alternative approach to assess the performance of dueling bandit algorithms. Here, we fix the number of human annotations (fixed annotation budget) and compute the accuracy in predicting the top-ranked system. As we show in figure 2, RMED achieves the highest top-rank prediction accuracy for any given number of human annotations. We provide the complete results in appendix F.1.

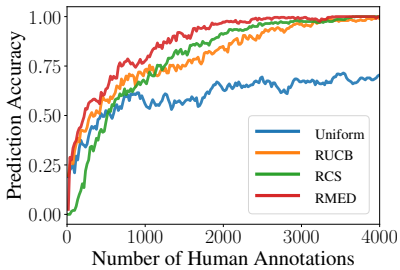

Figure 2: Top-rank prediction accuracy v/s number of human annotations used on WMT 16 tur-eng dataset

6.2 Performance of Evaluation Metrics

Before we utilize automatic evaluation metrics using our proposed model-based algorithms, we analyze the effectiveness of these metrics for pairwise NLG evaluations. In table 4, we report the sentence-level accuracy in predicting the comparison outcome w using direct assessment metrics with the Linear probability model (as discussed in section 3) along with our trained Electra metric. Across the tasks, we observe that metrics that utilize con-

Model-based Algorithm	Evaluation Metric	WMT 2016				WMT 2015			Grammarly		CoNLL '14 Task	E2E NLG	Para-Bank
		tur-eng	ron-eng	cze-eng	deu-eng	fin-eng	rus-eng	deu-eng	FCE	Wiki			
None (Model free)	None	2028	5113	1612	864	1707	1929	4047	2093	5647	9364	3753	24132
Random Mixing	Bleurt	237	1222	315	161	275	304	771	406	671	9584	1151	15874
	Electra	728	3213	385	152	236	512	650	1529	237	3302	326	1044
Uncertainty-aware Selection (STD)	Bleurt	103	1012	192	84	204	239	530	270	185	9356	1291	22876
	Electra	978	7251	478	210	388	962	1259	477	234	4708	199	2137
Uncertainty-aware Selection (BALD)	Bleurt	101	653	**136**	48	181	162	405	204	128	9356	1167	22619
	Electra	737	1648	223	114	207	538	488	281	75	1557	67	**858**
UCB Elimination	Bleurt	711	2684	1131	573	419	843	3556	967	1115	8382	2005	14098
	Electra	264	649	1131	414	294	1126	3556	3970	1115	2943	1112	9870
Uncertainty (BALD) + UCB Elim.	Bleurt	**31**	**415**	376	**25**	**59**	**82**	305	**162**	**39**	9995	256	4570
	Electra	721	736	144	51	76	288	**280**	312	45	**782**	**40**	2247

Table 3: Annotation complexity of model-based algorithms when used with RMED and Bleurt/Electra metric.

Metric	WMT (Avg.)	Gramm. (Avg.)	CoNLL '14 Task	E2E NLG	Para-Bank	TL; DR
Chrf	62.6	75.7	78.4	47.4	66.1	34.2
Bleu	41.5	73.2	78.9	45.0	63.8	42.8
Rouge-L	60.7	73.5	78.0	44.6	64.3	43.3
Embed. Avg.	56.5	70.1	76.0	49.8	64.9	38.2
Greedy Match.	59.5	68.1	77.7	46.5	64.7	43.1
Vector Extr.	59.4	66.0	76.3	44.9	63.7	47.4
BertScore	65.9	**77.4**	**82.0**	45.9	68.1	44.5
Laser	65.3	75.1	78.0	47.2	67.0	35.4
MoverScore	66.1	74.7	80.6	50.1	68.0	40.7
Bleurt	**68.2**	77.1	81.5	48.1	67.7	42.5
Electra (Ours)	65.7	74.0	81.6	**54.3**	**81.7**	-

Table 4: Sentence-level accuracy of direct assessment metrics with linear probability model and our trained Electra metric in predicting the comparison outcome

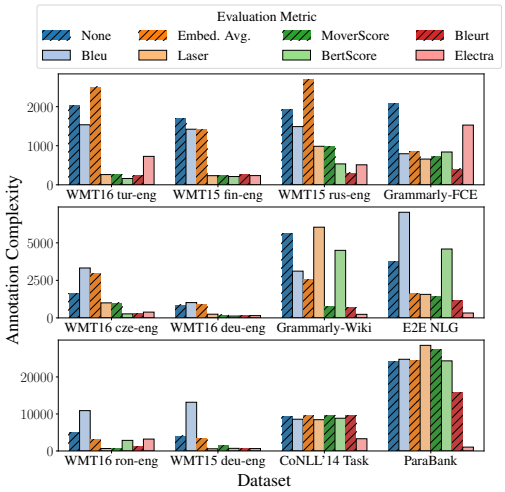

Figure 3: Annotation complexity of Random Mixing with RMED using various automatic evaluation metrics

textualized word embeddings, such as BertScore, perform much better than n-gram and static word embedding-based metrics. In MT, we observe that Bleurt, specifically finetuned on WMT human judgment data, performs the best. In Data-to-Text and Paraphrase generation, our trained Electra metric finetuned on task-specific data significantly outperforms the existing metrics. Interestingly, on the summarization task, all the existing metrics perform much worse than random predictions. Since *i.e.* they do not add any useful value in evaluation. Hence, we exclude the TLDR dataset from our analysis on model-based algorithms. Finally, as we show in appendix F.2, we observed that the performance is largely similar across all the three probability models: Linear, BTL, and BTL-logistic.

6.3 Analysis of Model-based Algorithms

We use our proposed model-based algorithms and incorporate the two best-performing evaluation metrics, *viz.*, Bleurt and Electra with the best performing dueling bandit algorithm, *viz.*, RMED. We compare the annotation complexity of various model-based algorithms in table 3. We observe that the Random Mixing algorithm with Bleurt and Electra reduces annotation complexity by 70.43% and 73.15%, respectively, when compared to the standard (model-free) RMED algorithm (row 1). Our Uncertainty-aware selection algorithm with the BALD measure further reduces the annotation complexity by around 37% (compared with Random Mixing). We notice that our UCB Elimination algorithm also provides significant improvements over standard RMED. Since UCB Elimination is complementary to Uncertainty-aware selection, we apply both these algorithms together and observe the lowest annotation complexity with a reduction of 89.54% using Electra and 84.00% using Bleurt over standard RMED. Lastly, in figure 3, we analyze the effect of using other evaluation metrics such as BLEU, BertSore, *etc.*, in Random Mixing. Interestingly, we notice that using metrics such as BLEU, which have low accuracy values, results in a higher annotation complexity than standard (model-free) RMED in some datasets. That is, we may even require a greater number of human annotations to over-compensate for the inaccu-

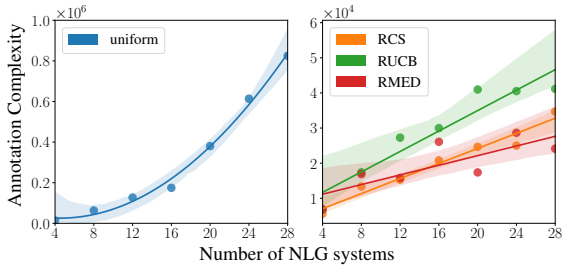

Figure 4: Annotation complexity of (model-free) uniform exploration and dueling bandit algorithms v/s the number of NLG systems on the ParaBank dataset

rate predictions from metrics like BLEU. However, with Laser, MoverScore, and BertScore, we observe significant reductions in annotation complexity. Please refer appendix F.3 for further results.

6.4 Effect of Number of NLG systems

We analyze how annotation complexity varies with the number of NLG systems. Specifically, we chose a subset of k systems out of the total 28 systems in the ParaBank dataset and computed the annotation complexity among these k systems. As shown in figure 4, the annotation complexity of uniform exploration grows quadratically with k as it explores all system pairs equally. However, for (model-free) dueling bandit algorithms such as RMED, the annotation complexity is much lower and only varies as $O(k)$. As shown in appendix F.4, we observed similar trends with model-based algorithms.

7 Practical Recommendations

We summarize the key insights from this study and provide practical recommendations on efficiently identifying the top-ranked NLG system.

1. Use RMED dueling bandit algorithm to actively choose system pairs for comparison.

2. If human evaluation datasets are available, train a metric to predict the comparison outcome directly. Otherwise, use Bleurt with any of the Linear, BTL, BTL-logistic models.

3. Manually annotate a few examples from the test dataset and evaluate the sentence-level accuracy of the metric. If the performance is poor (e.g., accuracy near the random baseline), do not use model-based approaches, obtain feedback only from human annotators.

4. If the metric is reasonably accurate, use UCB Elimination with Uncertainty-aware Selection (BALD). Tune the hyperparameters of these algorithms, if possible. Otherwise, refer appendix D for best practices developed based on analyzing the sensitivity of model-based algorithms to hyperparameters.

5. We can reduce the annotation time if we use multiple annotators in parallel. We observed that dueling bandit algorithms, though originally proposed for sequential annotations, are robust to asynchronous feedback from multiple annotators (Refer appendix E for details).

8 Related Work

Several works (Bojar et al., 2014, 2015; Sakaguchi et al., 2014, 2016) in Machine translation and Grammatical Error Correction adopt the TrueSkill algorithm (Herbrich et al., 2006), originally used for ranking Xbox gamers, to efficiently rank NLG systems from pairwise annotations. A recent work (Sakaguchi and Durme, 2018) proposes an online algorithm to rank NLG systems when we receive pairwise preference feedback in the form of a continuous scalar with bounded support. The key difference in our work is that we focus on the problem of identifying the top-rank system instead of ranking all the systems. Experimental study of dueling bandit algorithms have been limited to synthetic simulations in a few works (Yue and Joachims, 2011; Urvoy et al., 2013). Most others (Zoghi et al., 2014b,a; Komiyama et al., 2015; Zoghi et al., 2015; Wu and Liu, 2016) focus on information retrieval applications that involve evaluating search retrieval algorithms (Radlinski et al., 2008). To the best of our knowledge, ours is the first work to extensively study the effectiveness of dueling bandit algorithms for NLG evaluation.

9 Conclusion & Future work

In this work, we focused on the problem of identifying the top-ranked NLG system with few pairwise annotations. We formulated this problem in an Active Evaluation framework and showed that dueling bandit algorithms can reduce the number of human annotations by 80%. We then proposed model-based algorithms to combine automatic metrics with human evaluations and showed that human annotations can be reduced further by 89%; thereby requiring only a few hundred human annotations to identify the top-ranked system. In future work, we would like to extend our analysis to the general problem of finding the top-k ranked systems.

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Trans. Assoc. Comput. Linguistics*, 7:597–610.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256.

Viktor Bengs, Róbert Busa-Fekete, Adil El Mesaoudi-Paul, and Eyke Hüllermeier. 2021. Preference-based online learning with dueling bandits: A survey. *J. Mach. Learn. Res.*, 22:7:1–7:108.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ales Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 12–58. The Association for Computer Linguistics.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin M. Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, pages 131–198. The Association for Computer Linguistics.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation, WMT@EMNLP 2015, 17-18 September 2015, Lisbon, Portugal*, pages 1–46. The Association for Computer Linguistics.

R. Bradley and M. E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Bhuwan Dhingra, Manaal Faruqui, Ankur P. Parikh, Ming-Wei Chang, Dipanjan Das, and William W. Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4884–4895. Association for Computational Linguistics.

Ondrej Dusek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The E2E NLG challenge. *Comput. Speech Lang.*, 59:123–156.

Desmond Elliott and Frank Keller. 2014. Comparing automatic evaluation measures for image description. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 452–457. The Association for Computer Linguistics.

Moein Falahatgar, Yi Hao, Alon Orlitsky, Venkatadheeraj Pichapati, and Vaishakh Ravindrakumar. 2017a. Maxing and ranking with few assumptions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 7060–7070.

Moein Falahatgar, Alon Orlitsky, Venkatadheeraj Pichapati, and Ananda Theertha Suresh. 2017b. Maximum selection and ranking under noisy comparisons. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1088–1096. PMLR.

Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. 2014. Bootstrapping dialog systems with word embeddings. In *NeurIPS, modern machine learning and natural language processing workshop*, volume 2.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1050–1059. JMLR.org.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192. PMLR.

Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. Trueskill^{tm}: A bayesian skill rating system. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 569–576. MIT Press.

Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning. *CoRR*, abs/1112.5745.

J. Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. 2019. PARABANK: monolingual bitext generation and sentential paraphrasing via lexically-constrained neural machine translation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6521–6528. AAAI Press.

M. Kendall. 1948. Rank correlation methods.

Junpei Komiyama, Junya Honda, Hisashi Kashima, and Hiroshi Nakagawa. 2015. Regret lower bound and optimal algorithm in dueling bandit problem. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 1141–1154. JMLR.org.

Ilia Kulikov, Alexander H. Miller, Kyunghyun Cho, and Jason Weston. 2019. Importance of search and evaluation strategies in neural dialogue modeling. In *Proceedings of the 12th International Conference on Natural Language Generation, INLG 2019, Tokyo, Japan, October 29 - November 1, 2019*, pages 76–87. Association for Computational Linguistics.

Margaret Li, Jason Weston, and Stephen Roller. 2019. ACUTE-EVAL: improved dialogue evaluation with optimized questions and multi-turn comparisons. *CoRR*, abs/1909.03087.

Weixin Liang, J. Zou, and Zhou Yu. 2020. Beyond user self-reported likert scale ratings: A comparison model for automatic dialog evaluation. *ArXiv*, abs/2005.10716.

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.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguistics*, 8:726–742.

R. Luce. 1979. Individual choice behavior: A theoretical analysis.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2017. Sequence effects in crowdsourced annotations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2860–2865. Association for Computational Linguistics.

Soheil Mohajer, Changho Suh, and Adel M. Elmahdy. 2017. Active learning for top-k rank aggregation from noisy comparisons. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2488–2497. PMLR.

Courtney Napoles, Maria Nadejde, and J. Tetreault. 2019. Enabling robust grammatical error correction in new domains: Data sets, metrics, and analyses. *Transactions of the Association for Computational Linguistics*, 7:551–566.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and J. Tetreault. 2015a. Ground truth for grammaticality correction metrics. In *ACL*.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and J. Tetreault. 2015b. Ground truth for grammaticality correction metrics. In *ACL*.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2014, Baltimore, Maryland, USA, June 26-27, 2014*, pages 1–14. ACL.

Jekaterina Novikova, Ondrej Dusek, 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, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2241–2252. Association for Computational Linguistics.

Jekaterina Novikova, Ondrej Dusek, and Verena Rieser. 2018. Rankme: Reliable human ratings for natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 72–78. Association for Computational Linguistics.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

R. Plackett. 1975. The analysis of permutations. *Journal of The Royal Statistical Society Series C-applied Statistics*, 24:193–202.

Maja Popovic. 2015. chrf: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation, WMT@EMNLP 2015, 17-18 September 2015, Lisbon, Portugal*, pages 392–395. The Association for Computer Linguistics.

Filip Radlinski, Madhu Kurup, and Thorsten Joachims. 2008. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, pages 43–52. ACM.

Vasile Rus and Mihai C. Lintean. 2012. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, BEA@NAACL-HLT 2012, June 7, 2012, Montréal, Canada*, pages 157–162. The Association for Computer Linguistics.

Ananya B. Sai, Mithun Das Gupta, Mitesh M. Khapra, and Mukundhan Srinivasan. 2019. Re-evaluating ADEM: A deeper look at scoring dialogue responses. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6220–6227. AAAI Press.

Ananya B. Sai, Akash Kumar Mohankumar, Siddhartha Arora, and Mitesh M. Khapra. 2020a. Improving dialog evaluation with a multi-reference adversarial dataset and large scale pretraining. *Trans. Assoc. Comput. Linguistics*, 8:810–827.

Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2020b. A survey of evaluation metrics used for NLG systems. *CoRR*, abs/2008.12009.

Keisuke Sakaguchi and Benjamin Van Durme. 2018. Efficient online scalar annotation with bounded support. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 208–218. Association for Computational Linguistics.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel R. Tetreault. 2016. Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Trans. Assoc. Comput. Linguistics*, 4:169–182.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 1–11. The Association for Computer Linguistics.

João Sedoc, Daphne Ippolito, Arun Kirubarajan, Jai Thirani, Lyle Ungar, and Chris Callison-Burch. 2019. Chateval: A tool for chatbot evaluation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 60–65. Association for Computational Linguistics.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1702–1723. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. BLEURT: learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7881–7892. Association for Computational Linguistics.

Edwin D. Simpson and Iryna Gurevych. 2018. Finding convincing arguments using scalable bayesian preference learning. *Trans. Assoc. Comput. Linguistics*, 6:357–371.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Nisan Stiennon, L. Ouyang, Jeff Wu, D. Ziegler, Ryan J. Lowe, Chelsea Voss, A. Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. *ArXiv*, abs/2009.01325.

Katsuhito Sudoh, Kosuke Takahashi, and Satoshi Nakamura. 2021. Is this translation error critical?: Classification-based human and automatic machine translation evaluation focusing on critical errors. In *HUMEVAL*.

Balázs Szörényi, Róbert Busa-Fekete, Adil Paul, and Eyke Hüllermeier. 2015. Online rank elicitation for plackett-luce: A dueling bandits approach. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 604–612.

Tanguy Urvoy, F. Clérot, R. Féraud, and Sami Naamane. 2013. Generic exploration and k-armed voting bandits. In *ICML*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. Tl;dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization, NFiS@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 59–63. Association for Computational Linguistics.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Huasen Wu and Xin Liu. 2016. Double thompson sampling for dueling bandits. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 649–657.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, abs/2010.11934.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. 2012. The k-armed dueling bandits problem. *J. Comput. Syst. Sci.*, 78(5):1538–1556.

Yisong Yue and Thorsten Joachims. 2011. Beat the mean bandit. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 241–248. Omnipress.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 563–578. Association for Computational Linguistics.

Masrour Zoghi, Zohar S. Karnin, Shimon Whiteson, and Maarten de Rijke. 2015. Copeland dueling bandits. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 307–315.

Masrour Zoghi, Shimon Whiteson, Maarten de Rijke, and Rémi Munos. 2014a. Relative confidence sampling for efficient on-line ranker evaluation. In *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, pages 73–82. ACM.

Masrour Zoghi, Shimon Whiteson, Rémi Munos, and Maarten de Rijke. 2014b. Relative upper confidence bound for the k-armed dueling bandit problem. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 10–18. JMLR.org.

A Further Details on Experiments

A.1 Tasks & Datasets

In table 5, we report the dataset statistics along with links to download the original datasets. We now discuss the preprocessing steps:

Machine Translation: In WMT 2015 and 2016 tasks, human annotators were asked to rank five system outputs (translated sentences) relative to each other. As recommended by the organizers (Bojar et al., 2014), we convert each of these rankings into $\binom{5}{2}$ pairwise comparisons of systems.

Grammatical Error Correction: The Grammarly evaluation datasets follow the RankME (Novikova et al., 2018) annotation style where annotators were shown 8 outputs side by side for each input and were asked to provide a numerical score to each of them. We discarded one of the outputs out of the 8, which was human crafted, and used the remaining 7 model-generated outputs. We then convert these 7 scores into $\binom{7}{2}$ pairwise comparisons of systems. Human evaluations of the CoNLL-2014 Shared Task followed the same process as WMT 2015. Hence, we follow the same preprocessing steps as WMT.

Data-to-Text Generation: The E2E NLG Challenge also follows the RankME annotation format. We follow the same preprocessing steps as the Grammarly datasets. Out of the total 21 systems, we held out 5 systems to train the Electra model and use the remaining 16 systems.

Paraphrase Generation: For ParaBank, we follow the same preprocessing steps as the Grammarly datasets. Out of the total 35 systems, we held out of 7 systems and only used the remaining 28 systems.

Summarization: We select 11 systems that have human annotations between each pair of them. These systems are GPT3-like models with varying model sizes (3B, 6B, 12B) and training strategies. We do not perform any additional preprocessing here.

A.2 Direct Assessment Metrics: Implementation Details

We use the nlg-eval library[2] for the implementation of BLEU-4, ROUGE-L, Embedding Average, Vector Extrema, and Greedy Matching. For chrF, Laser and BertScore, we use the implementations from the VizSeq library [3]. We use the official implementation released by the original authors for MoverScore and Bleurt. Among these metrics, Bleurt is the only trainable metric. We use the publicly released Bleurt-base checkpoint trained on WMT direct judgments data. As described in section 4.2, we apply Dropout to the Bleurt model during test time to estimate prediction uncertainty.

[2] https://github.com/Maluuba/nlg-eval

[3] https://github.com/facebookresearch/vizseq

A.3 Finetuning Datasets

Here, we describe the task-specific datasets used for finetuning the Electra model (pairwise evaluation metric described in section 5.2). For MT, we used human evaluations of WMT 2013 and 2014, consisting of a total of 650k examples. For GEC, we curated a training dataset of 180k pairs of texts and human preference using data released by (Napoles et al., 2015b) and the development set released by (Napoles et al., 2019). We utilize 11k examples from 5 held-out systems in the E2E NLG Challenge (apart from the 16 systems used for evaluations) for Data-to-Text generation. Lastly, we use a dataset of 180k examples from 7 held-out systems in the ParaBank dataset for paraphrase generation. We use $90\% - 10\%$ split for splitting the dataset into train and validation sets. Note that these datasets do not have any overlap with the datasets used for evaluating dueling bandit algorithms.

A.4 Finetuning Details

We use the pretrained Electra-base model (Clark et al., 2020) with 110M parameters (12 layers and 12 attention heads) as our base model. We finetune the model using ADAM optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We use a linear learning rate decay with a maximum learning rate of 1e-5 and warm-up for 10% of training. We use a batch size of 128 and finetune for four epochs. We finetune all the models on Google Cloud TPU v3-8. To estimate prediction, we apply Dropout to the Electra model during test time as described in 4.2.

B Summary of Dueling Bandit Algorithms

We now provide an overview of various dueling bandit algorithms in the literature. We first introduce a few additional notations and terminologies in B.1. Later in B.2, we describe the various structural assumptions made by different dueling bandit algorithms. Finally, in B.3, we summarize 13 dueling bandit algorithms that we analyze in this work.

Task	Dataset	# Systems	# Human Annotations	Label Distrib. (0-0.5-1)	Downloadable Link
Machine Translation	WMT15 fin-eng	14	31577	37%-26%-37%	Click here
	WMT15 rus-eng	13	44539	36%-27%-37%	
	WMT15 deu-eng	13	40535	32%-36%-32%	
	WMT16 tur-eng	9	10188	28%-44%-28%	Click here
	WMT16 ron-eng	7	15822	38%-24%-38%	
	WMT16 cze-eng	12	125788	38%-25%-37%	
	WMT16 deu-eng	10	20937	37%-26%-37%	
Grammatical Error Correction	Grammarly (FCE)	7	20328	29%-40%-31%	Click here
	Grammarly (Wiki)	7	20832	29%-40%-31%	
	CoNLL-2014 Shared Task	13	16209	23%-52%-25%	Click here
Data-to-Text Generation	E2E NLG Challenge	16	17089	24%-50%-26%	Click here
Paraphrase Generation	ParaBank	28	151148	44%-2%-54%	Click here
Summarization	TLDR OpenAI	11	4809	49%-0%-51%	Click here

Table 5: Description of tasks and datasets with the number of NLG systems, number of pairwise human annotations, label distribution and the downloadable links to the datasets before preprocessing

B.1 Notations and Terminologies

Let $\Delta_{ij} = p_{ij} - \frac{1}{2}$ where p_{ij} is the preference probability of system i over j, as defined in section 2.3. We call a system as the Copeland winner if it beats more number of systems than any other system. Mathematically, a Copeland winner i^* is defined as $i^* = \arg\max_i \sum_{j=1}^{k} \mathbb{1}(\Delta_{ij} > 0)$. A special case of the Copeland winner is the Condorcet winner, which is the system that beats all other systems. In all our NLG tasks and datasets, we observed that this special case holds true *i.e.* there exists a system that beats all other $k-1$ systems, and we define it as the top-ranked system. Nevertheless, we mention these two definitions to distinguish algorithms that work for the general Copeland winner, even if the Condorcet winner does not exist.

B.2 Assumptions

All the dueling bandit algorithms that we analyze in this work assume a stochastic feedback setup in which the feedback is generated according to an underlying (unknown) stationary probabilistic process. Specifically, in our Active Evaluation framework, this is equivalent to assuming that the annotator preference is stationary over time and is given by some fixed distribution $p_a(w|Y_1^{(t)}, Y_2^{(t)})$. Further, many dueling bandit algorithms make various assumptions on the true pairwise preferences and exploit these assumptions to derive theoretical guarantees (Bengs et al., 2021). In table 6, we describe the various commonly used assumptions by dueling bandit algorithms. For example, the stochastic triangle inequality assumption (STI), described in row 4 of table 6, assumes that the true preference probabilities between systems obey the triangle inequality. We note here that one cannot verify the validity of these assumptions apriori since we do not have access to the true preferences.

B.3 Algorithms

In table 7, we describe the various dueling bandit algorithms along with the assumptions (used to provide theoretical guarantees) and the target winner. We summarize these algorithms below:

IF: Interleaved Filtering (IF) (Yue et al., 2012) algorithm consists of a sequential elimination strategy where a currently selected system s_i is compared against the rest of the active systems (not yet eliminated). If the system s_j beats a system s_i with high confidence, then s_i is eliminated, and s_j is compared against all other active systems. Similarly, if the system s_i beats s_j with high confidence, then s_j is eliminated, and s_i is continued to be compared against the remaining active systems. Under the assumptions of TO, SST, and STI, the authors provide theoretical guarantees for the expected regret achieved by IF.

BTM: Beat The Mean (BTM) (Yue and Joachims, 2011), similar to IF, is an elimination-based algorithm that selects the system s_i with the fewest comparisons and compares it with a randomly chosen system from the set of active systems. Based on the comparison outcome, a score and confidence interval are assigned to the system s_i. BTM eliminates a system as soon as there is another system with a significantly higher score.

Knockout, Seq Elim, Single Elim: Knockout (Falahatgar et al., 2017b), Sequential Elimination (Falahatgar et al., 2017a), Single Elimination (Mohajer et al., 2017) are all algorithms that proceed in a knockout tournament fashion where the systems are randomly paired, and the winner in each duel will play the next round (losers are knocked out) until the overall winner is determined. During a duel, the algorithm repeatedly compares the two systems to reliably determine the winner. The key difference between the three algorithms is the assumptions they use and how they determine the number of comparisons required to identify the winning system in a duel with high probability.

Plackett Luce: Plackett Luce Condorcet winner identification algorithm (Szörényi et al., 2015) assumes that the true rank distribution follows the Placket-Luce model (Plackett, 1975). The algorithm is based on a budgeted version of QuickSort. The authors show that it achieves a worst-time annotation complexity of the order $k \log k$ under the Placket-Luce assumption.

RUCB: Relative Upper Confidence Bound (RUCB) (Zoghi et al., 2014b) is an adaptation of the well-known UCB algorithm (Auer et al., 2002) to the dueling bandit setup. Similar to UCB, RUCB selects the first system $s_t^{(1)}$ based on "optimistic" estimates of the pairwise preference probabilities *i.e.* based on an upper confidence bound of preference probabilities. The second system $s_t^{(2)}$ is chosen to be the one that is most likely to beat $s_t^{(1)}$.

RCS: Relative Confidence Sampling (RCS) (Zoghi et al., 2014a) follows a Bayesian approach by maintaining a posterior distribution over the preference probabilities. At each time step t, the algorithm samples preference probabilities from the posterior and simulates a round-robin tournament among the systems to determine the Condorcet winner. The estimated Condorcet winner is chosen as the first system $s_t^{(1)}$ and second system $s_t^{(2)}$ is chosen such that it has the best chance of beating $s_t^{(1)}$.

RMED: Relative Minimum Empirical Divergence1 (RMED) algorithm (Komiyama et al., 2015) maintains an empirical estimate of the "likelihood" that a system is the Condorcet winner. It then uses this estimate to sample the first system $s_t^{(1)}$ and then selects the second system $s_t^{(2)}$ that is most likely to beat $s_t^{(1)}$.

SAVAGE: Sensitivity Analysis of VAriables for Generic Exploration (SAVAGE) (Urvoy et al., 2013) is a generic algorithm that can be adopted for various ranking problems such as Copeland winner identification. SAVAGE (Copeland) algorithm, at each time step, randomly samples a pair of systems from the set of active system pairs (not yet eliminated) and updates the preference estimates. A system pairs (s_i, s_j) is eliminated if either (i) the result of comparison between s_i and s_j is already known with high probability, or (ii) there exists some system s_k where the estimated Copeland score of s_k is significantly higher than s_i or s_j.

CCB: Copeland Confidence Bound (CCB) (Zoghi et al., 2015) is similar to the RUCB algorithm but is designed to identify the Copeland Winner (a generalization of the Condorcet winner). The CCB algorithm maintains optimistic preference estimates and uses them to choose the first system $s_t^{(1)}$ and then selects the second system $s_t^{(2)}$ that is likely to discredit the hypothesis that $s_t^{(1)}$ is indeed the Copeland winner. The algorithm successively removes all other systems that are highly unlikely to be a Copeland winner.

Assumption Name	Condition
Total Order (TO)	$\exists$ a total order $\succ$ over $\mathcal{S}$: $i \succ j \iff \Delta_{ij} > 0$
Strong stochastic transitivity (SST)	$\Delta_{ij} > 0, \Delta_{jk} > 0 \implies \Delta_{ik} \geq \max(\Delta_{ij}, \Delta_{jk})$
Relaxed stochastic transitivity (RST)	$\exists \gamma \geq 1$: $\Delta_{ij} > 0, \Delta_{jk} > 0 \implies \gamma\Delta_{ik} \geq \max(\Delta_{ij}, \Delta_{jk})$
Stochastic triangle inequality (STI)	$\Delta_{ij} > 0, \Delta_{jk} > 0 \implies \Delta_{ik} \leq \Delta_{ij} + \Delta_{jk}$
Condorcet winner (CW)	$\exists i^*$: $\Delta_{i^*,j} > 0, \forall j \in \mathcal{S} - i^*$
PL model	The underlying rank distribution follows the Plackett-Luce (PL) model (Plackett, 1975; Luce, 1979)

Table 6: Various assumptions made by dueling bandit algorithms in the literature

Algorithm	Assumptions	Target
IF (Yue et al., 2012)	TO+SST+STI	Condorcet
BTM (Yue and Joachims, 2011)	TO+RST+STI	Condorcet
Seq-Elim. (Falahatgar et al., 2017a)	SST	Condorcet
Plackett Luce (Szörényi et al., 2015)	PL model	Condorcet
Knockout (Falahatgar et al., 2017b)	SST+STI	Condorcet
Single Elim.(Mohajer et al., 2017)	TO	Condorcet
RUCB (Zoghi et al., 2014b)	CW	Condorcet
RCS (Zoghi et al., 2014a)	CW	Condorcet
RMED (Komiyama et al., 2015)	CW	Condorcet
SAVAGE (Urvoy et al., 2013)	-	Copeland
CCB (Zoghi et al., 2015)	-	Copeland
DTS (Wu and Liu, 2016)	-	Copeland
DTS++ (Wu and Liu, 2016)	-	Copeland

Table 7: Summary of dueling bandits algorithms in the literature along with their theoretical assumptions and the target winner of the learner

DTS, DTS++: The Double Thompson Sampling (DTS) algorithm (Wu and Liu, 2016) maintains a posterior distribution over the pairwise preference matrix, and selects the system pairs $s_t^{(1)}, s_t^{(2)}$ based on two independent samples from the posterior distribution. The algorithm updates the posterior distributions based on the comparison outcome and eliminates systems that are unlikely to be the Copeland winner. DTS++ is an improvement proposed by the authors, which differs from DTS in the way the algorithm breaks ties. Both have the same theoretical guarantees, but DTS++ has been empirically shown to achieve better performance (in terms of regret minimization).

C Hyperparameters Details

We discuss the details of the hyperparameters and the tuning procedure used for dueling bandit algorithm in C.1, pairwise probability models in C.2 and our model-based algorithm in C.3. In all three cases, we use the validation split of the finetuning datasets described in A.3 as our validation dataset. For example, the validation split of the finetuning datasets for MT consists of 10% of the WMT 2013 and 2014 datasets. We use this dataset to tune the hyperparameters for WMT 2015 and 2016 datasets.

C.1 Dueling Bandit Algorithms

For all algorithms other than Knockout and Single Elimination, we use the hyperparameters recommended by the original authors for all the datasets. For example, in the RMED algorithm, described in algorithm 1 of (Komiyama et al., 2015), we use $f(K) = 0.3K^{1.01}$ as suggested by the authors. For the RCS algorithm, described in algorithm 1 of (Zoghi et al., 2014a), we use α (exploratory constant) $= 0.501$. For RUCB (algorithm 1 of (Zoghi et al., 2014b)), we use $\alpha = 0.51$. Similarly, for all algorithms other than Knockout and Single Elimination, we use the recommended hyperparameters mentioned in the original paper. For knockout and Single Elimination, we found that the performance was very sensitive to the hyperparameters. For these two algorithms, we manually tuned the hyperparameters on the validation set. In Knockout, algorithm 3 of (Falahatgar et al., 2017b), we use $\epsilon = 0.2, \delta = 0.05, \gamma = 1.0$ for WMT'16 ron-eng and TLDR OpenAI datasets. We use $\epsilon = 0.2, \delta = 0.05, \gamma = 0.6$ for ParaBank and Grammarly-Wiki datasets and $\epsilon = 0.2, \delta = 0.09, \gamma = 0.6$ for all other datasets. In Single Elimination, we use m (number of pairwise comparisons per duel) $= 1000$ for WMT'16 ron-eng, E2E NLG, Grammarly-FCE, $m = 1500$ for CoNLL'14 shared task and $m = 500$ for all other datasets.

C.2 Pairwise Probability Models

Let $\tilde{f}(Y)$ be the unnormalized score given an automatic evaluation metric for an hypothesis Y. We preprocess the score $\tilde{f}(Y)$ to obtain $f(Y)$ to ensure that the pairwise probability scores is always a valid *i.e.* lies between 0 and 1. To preprocess the scores, we use the validation dataset consisting of tuples of the form $\{Y_1^{(i)}, Y_2^{(i)}, w^{(i)}\}_{i=1}^N$ where $Y_1^{(i)}$, $Y_2^{(i)}$ represent the ith generated texts and $w^{(i)}$ is the corresponding comparison outcome provided by human annotators.

Linear: Let $\Delta_i = |\tilde{f}(Y_1^{(i)}) - \tilde{f}(Y_2^{(i)})|$ and $\Delta = \max_i \Delta_i$. We divide the unormalized $\tilde{f}(Y)$ scores by 2Δ *i.e.*

$$f(Y) = \frac{\tilde{f}(Y)}{2\Delta}$$

.

BTL: Let $f_i^m = \max\{\tilde{f}(Y_1^{(i)}), \tilde{f}(Y_2^{(i)})\}$, $f^m = \max_i f_i^m$. We now subtract the scores by f^m to ensure that the scores are non-negative *i.e.*

$$f(Y) = \tilde{f}(Y) - f^m$$

BTL-Logistic: BTL-Logistic model always provides a score between 0 and 1. However, we found that dividing the scores by a temperature co-efficient γ can provide better results *i.e.*

$$f(Y) = \frac{\tilde{f}(Y)}{\gamma}$$

We tune γ using grid search between 0.005 and 1 on the validation set to minimize the cross-entropy loss between the preference probabilities $\hat{p}(Y_1 \succ Y_2)$ and the human labels w.

Thresholds: As described in section 3, we threshold the preference probabilities $\hat{p}(Y_1 \succ Y_2)$ at two thresholds τ_1 and τ_2 to obtain the predicted comparison outcome $\hat{w}$. We perform a grid search by varying τ_1 from 0.4 to 0.5 and τ_2 from 0.5 to 0.6 with a step size of 0.001. We choose the optimal thresholds that maximize the prediction accuracy on the validation dataset.

Dataset	Rand. Mix.	Uncertainty (BALD)	UCB-Elim.	
	p_m	τ_{BALD}	α	τ_{cop}
WMT (all 7 datasets)	0.8	0.025	0.5	0.8
Grammarly (FCE & Wiki)	0.8	0.07	0.5	0.8
CoNLL'14	0.8	0.07	0.5	0.8
E2E NLG	0.9	0.035	0.5	0.8
ParaBank	0.95	0.15	0.5	0.8

Table 8: Tuned Hyperparameters of Model-based algorithms when used with the Electra Metric

Dataset	Rand. Mix.	Uncertainty (BALD)	UCB-Elim.	
	p_m	τ_{BALD}	α	τ_{cop}
WMT (all 7 datasets)	0.8	0.005	0.5	0.8
Grammarly (FCE & Wiki)	0.8	0.0005	0.5	0.8
CoNLL'14	0.01	0.00005	1	0.7
E2E NLG	0.7	0.0025	0.5	0.8
ParaBank	0.4	0.0005	0.5	0.8

Table 9: Tuned Hyperparameters of Model-based algorithms when used with the Bleurt Metric

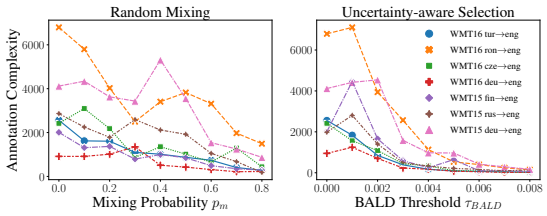

Figure 5: Variation in annotation complexity with Mixing probability in Random Mixing with Bleurt on the left and with BALD threshold in Uncertainty-aware Selection (BALD) with Bleurt on the right

C.3 Model-based Algorithms

We manually tune the hyperparameters in our model-based algorithms on the validation dataset. For clarity, we first describe the hyperparameters in the different model-based algorithms. In Random Mixing, we need to choose the mixing probability p_m hyperparameter. In Uncertainty-aware Selection (BALD), we need to choose a threshold value τ_{BALD} for the BALD score at which we decide to ask for human annotations. For UCB elimination, we should choose a threshold τ_{cop} for optimistic Copeland scores and the α hyperparameter, which controls the size of the confidence region. In table 8 and 9, we report the tuned hyperparameter values when using Electra and Bleurt (with the Linear probability model) as the evaluation model. Another hyperparameter is the number of Monte-Carlo samples L to obtain from the Dropout distribution as discussed in section 4.2. We set $L = 20$, *i.e.* we independently apply dropout 20 times for each test predictions.

D Effect of Hyperparameters in Model-based Algorithms

D.1 Sensitivity to Hyperparameters

We study how hyperparameters in our proposed model-based algorithms affect annotation complexity. Recall that in Random Mixing, the mixing probability p_m controls the ratio of real and model generated feedback given to the learner. In Uncertainty-aware Selection (BALD), we obtain human annotations when the BALD score is above a threshold τ_{BALD}. Here, as well τ_{BALD} implicitly controls the fraction of real and predicted feedback. In figure 5, we show the effect of p_m in Random Mixing with Bleurt and τ_{BALD} in Uncertainty-aware Selection with Bleurt. We observe that with increases in both the hyperparameters, the annotation complexity decreases, *i.e.*, with a greater amount of feedback received from Bleurt, the number of required human annotations is lower. However, as shown in figure 6, we observe the opposite trend when we use metrics such as BLEU, which are highly inaccurate. In these cases, we require a greater number of human annotations to compensate for the highly erroneous feedback received from the evaluation metric. Therefore, the optimal mixing probability p_m in such cases is close to 0 *i.e.* equivalent to the model-free case. For moderately accurate metrics such as Laser, we observed the optimal p_m was close to 0.4 to 0.6. The key insight from these observations is that the higher the accuracy of the metric, the higher amount of feedback can be obtained from the metric to identify the top-ranked system. In figure 7, we analyze how the annotation complexity of UCB Elimination with Bleurt varies with the optimistic Copeland threshold τ_{cop} hyperparameter. We fixed α hyperparameter to 0.6. We observed that UCB Elimination is much more robust to τ_{cop} and a general value of $\tau_{cop} = 0.8$ worked well across all datasets and metrics.

D.2 Best Practices in Choosing Hyperparameters

The optimal approach to choose hyperparameters is usually to tune them on a validation set. But, at

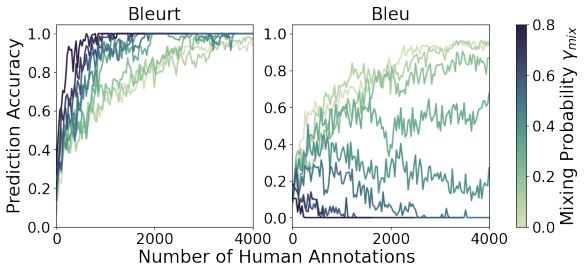

Figure 6: Prediction accuracy v/s number of human annotations collected for Random Mixing with Bluert and BLEU for different mixing probability p_m on the WMT 15 deu-eng dataset

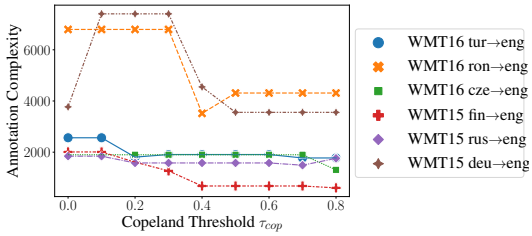

Figure 7: Annotation complexity of UCB Elimination with Bleurt v/s the Copland threshold for $\alpha = 0.6$

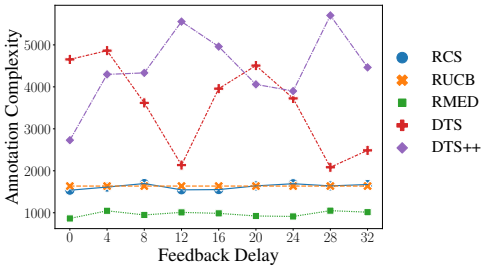

Figure 8: Annotation Complexity v/s delays in feedback on the WMT16 deu-eng dataset

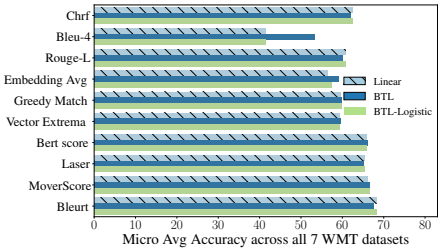

Figure 9: Sentence-level prediction accuracy of direct assessment metrics with the Linear, BTL, and BTL-Logistic models averaged across the 7 WMT datasets

times, it may not be possible either because of computational reasons or because a human-annotated validation dataset may not be available. In such cases, we provide a few heuristics based on our previous analysis to choose hyperparameters in our model-based algorithms:

1. Choose the mixing probability p_m in Random Mixing proportionately with the accuracy of the metric. For example, we observed that for metrics with sentence-level prediction accuracy greater than 70%, $p_m = 0.8$ tend to work well. For accuracy between 65% to 70%, p_m in the range of 0.5-0.7 worked well.

2. Once we choose a value of p_m, we can find an appropriate BALD threshold τ_{BALD} where $100 \times p_m\%$ of BALD scores are above τ_{BALD} and $100 \times (1-p_m)\%$ of BALD score are below τ_{BALD}. Choosing the BALD threshold this way ensures that we can directly control the desired amount of model-predicted feedback given to the learner.

3. For UCB Elimination, we recommend using the default values of $\alpha = 0.6$ and $\tau_{cop} = 0.8$, which we found to work well across tasks and metrics.

E Robustness to Delayed Feedback

In some instances, human annotations are obtained from multiple crowdsourced annotators in parallel to reduce the time taken for annotations. In such cases, the learner is required to choose the system pairs $(s_1^{(t)}, s_2^{(t)})$ to give to some annotator i even before we obtain the result $w^{(t-1)}$ of the previous comparison from some other annotator j. In other words, the learner may experience a delay $d > 0$ in feedback where at time t, the learner may only have access to the comparison history up to time $t-d-1$. As shown in figure 8, we observe that the top-performing dueling bandit algorithms tend to be robust to delays in feedback. We notice that the variation in the annotation complexity of RMED and RCS as measured by standard deviation is only 64.49 and 62.86, respectively.

F Additional Results

F.1 Results of Dueling Bandit Algorithms

We report the annotation complexity of all 13 dueling bandit algorithms on 13 evaluation datasets in table 10. In figure 10, we show the top-rank prediction accuracy as a function of the number of human annotations for various dueling bandit algorithms on all the datasets, other than WMT 16 tur-eng, which is separately depicted in figure 2.

Algorithm	WMT 2016				WMT 2015			Grammarly		CoNLL	E2E	Para-	TL;
	tur-eng	ron-eng	cze-eng	deu-eng	fin-eng	rus-eng	deu-eng	FCE	Wiki	'14 Task	NLG	Bank	DR
Uniform	19479	24647	10262	3032	2837	12265	17795	8115	34443	61369	65739	825211	5893
IF	117762	282142	135718	75014	101380	162536	261300	226625	364304	713522	718492	605825	70071
BTM	32010	17456	$> 10^6$	2249	2926	11108	8328	2778	$> 10^6$	$> 10^6$	**2541**	10175	2038
Seq-Elim.	10824	17514	5899	4440	16590	6881	17937	12851	48068	38554	41037	$> 10^6$	9046
PL	7011	18513	4774	4618	7859	17049	15215	8037	13156	**5682**	60031	$> 10^6$	3871
Knockout	3415	7889	4723	3444	5104	5809	5956	3134	3777	8055	7708	17418	4953
Sing. Elim.	4830	6000	5885	5340	6953	6465	6453	6000	9000	12940	15000	55900	9045
RUCB	3125	5697	3329	1636	**1655**	4536	6222	2732	5617	19024	10924	41149	1647
RCS	2442	**3924**	3370	1537	2662	3867	5296	**1816**	**4606**	12678	7263	34709	1903
RMED	**2028**	5113	**1612**	**864**	1707	**1929**	**4047**	2093	5647	9364	3753	**24132**	**1162**
SAVAGE	10289	18016	6639	2393	2675	12806	12115	5767	22959	39208	41493	255208	4733
CCB	7017	11267	5389	2884	4092	11548	10905	4386	10020	21392	16960	87138	2518
DTS	10089	9214	8618	4654	4850	13317	16473	4355	11530	18199	19940	170467	1354
DTS++	7626	9483	5532	2729	6465	9394	14926	9284	17774	31562	15065	52606	6284

Table 10: Annotation complexity of 13 dueling bandit algorithms along with the uniform exploration algorithm on 13 datasets spanning 5 NLG tasks

Metrics	WMT (Micro Average)			Grammarly (Micro Average)			CoNLL-2014 Shared Task			E2E NLG Challenge			ParaBank			TLDR OpenAI		
	Linear	BTL	BTL Log.	Linear	BTL	BTL Log.	Linear	BTL	BTL Log.	Linear	BTL	BTL Log.	Linear	BTL	BTL Log.	Linear	BTL	BTL Log.
Chrf	62.6	62.0	62.6	75.7	75.3	75.9	78.4	78.3	78.4	47.4	48.8	48.3	66.1	66.1	66.1	34.2	35.4	35.4
Bleu-4	41.5	53.4	41.5	73.2	73.0	73.2	78.9	78.7	78.9	45.0	39.0	50.1	63.8	63.2	63.8	42.8	44.0	42.8
Rouge-L	60.7	60.0	60.7	73.5	73.6	73.6	78.0	78.0	78.0	44.6	43.8	50.2	64.3	64.3	64.3	43.3	43.3	43.3
Emb. Avg.	56.5	59.1	57.5	70.1	70.3	71.5	76.0	76.7	77.0	49.8	51.6	51.8	64.9	64.9	64.9	38.2	38.2	38.2
Greedy Match	59.5	59.8	59.9	68.1	68.4	68.2	77.7	77.4	77.7	46.5	48.8	48.9	64.7	64.7	64.5	43.1	43.1	43.1
Vector Extr	59.4	59.5	59.3	66.0	66.9	66.5	76.3	76.7	76.7	44.9	46.2	49.1	63.7	63.7	63.7	47.4	47.1	48.1
Bertscore	65.9	66.2	65.9	77.4	77.2	77.4	82.0	81.5	82.0	45.9	49.3	50.1	68.1	68.1	68.1	44.5	44.4	44.5
Laser	65.3	65.1	65.3	75.1	73.0	75.1	78.0	76.4	78.0	47.2	49.9	50.5	67.0	67.0	67.0	35.4	35.4	35.4
MoverScore	66.1	66.5	66.1	74.7	70.9	73.0	80.6	79.6	80.3	50.1	49.3	50.4	68.0	68.0	67.8	40.7	40.7	40.7
Bleurt	68.2	67.5	68.2	77.1	76.6	76.0	81.5	81.5	80.8	48.1	50.4	50.4	67.7	67.7	67.7	42.5	42.5	42.3
Electra	65.7			74.0			81.6			54.3			81.7			-		

Table 11: Sentence-level accuracy of direct assessment metrics with linear, BTL, and BTL-logistic probability models and our trained Electra metric in predicting the comparison outcome

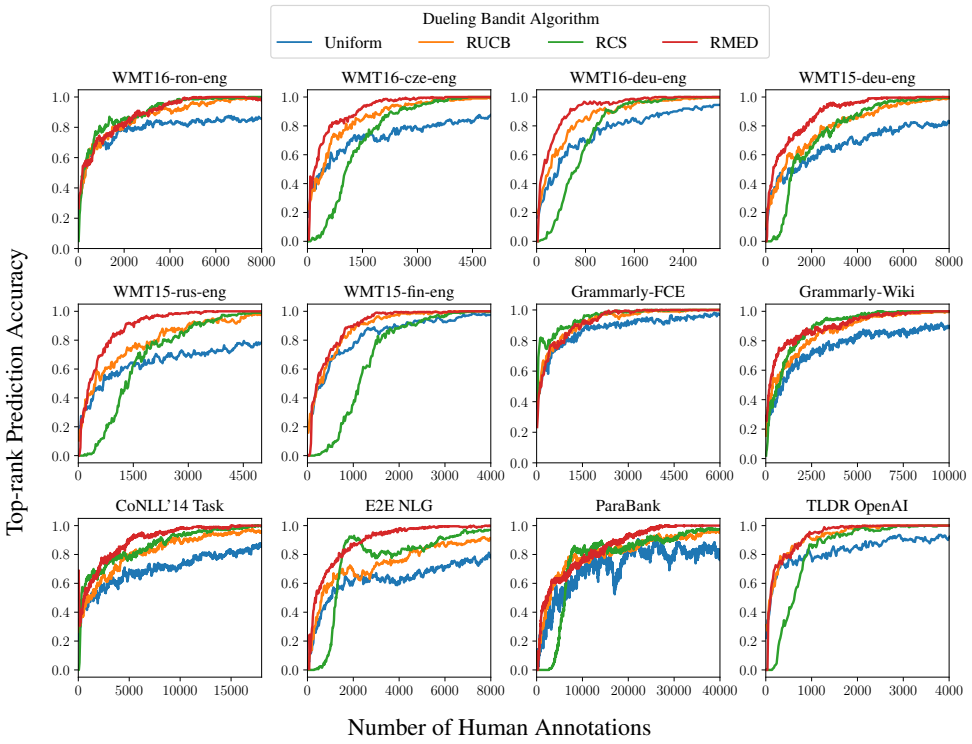

Figure 10: Top-rank prediction accuracy as a function of the number of human annotations for (model-free) Uniform exploration and RUCB, RCS, and RMED dueling bandit algorithms on 12 NLG datasets

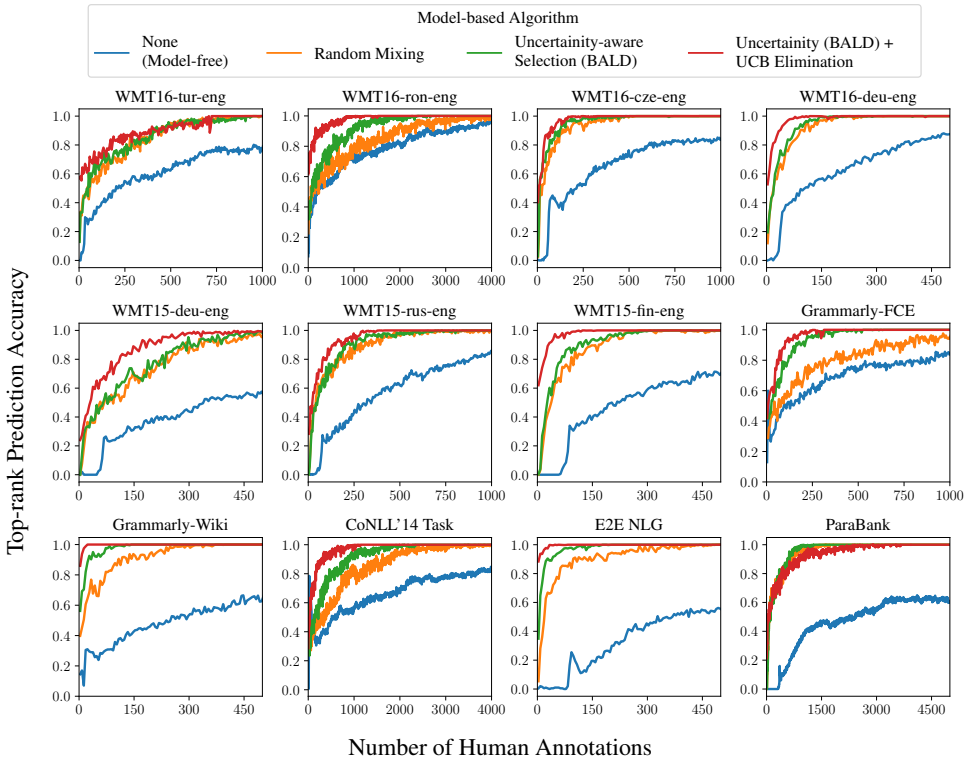

Figure 11: Top-rank prediction accuracy as a function of the number of human annotations for various model-based dueling bandit algorithms with RMED and Electra metric on 12 NLG datasets

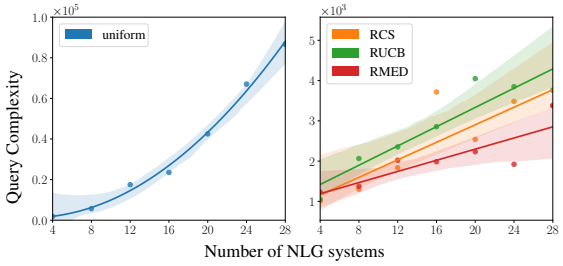

Figure 12: Annotation complexity of Random Mixing using the Electra metric with uniform exploration and dueling bandit algorithms as function of number of NLG systems on the ParaBank dataset

F.2 Performance of Evaluation Metrics

In table 11, we report the sentence-level accuracy in predicting the comparison outcome for 10 direct assessment metrics using three probability models along with the trained pairwise metric (Electra). We observe that there is little variation in performance across the three probability models. To further illustrate this, we plot the accuracy on the WMT datasets in figure 9 and observe that the performance is largely similar across Linear, BTL, and BTL-logistic models.

F.3 Model-based Algorithms

In figure 11, we show the top-rank prediction accuracy as a function of the number of human annotations for various model-based algorithms using the Electra metric with RMED. We observe that Random Mixing and Uncertainty-aware Selection (BALD) algorithms have significantly higher prediction accuracy than model-free RMED for any given number of human annotations. Further, when we use UCB Elimination with Uncertainty-aware Selection, we observe the highest top-rank prediction accuracy for any given number of annotations.

F.4 Effect of number of NLG systems

In figure 12, we compare the variations in annotation complexity of Random Mixing (with Electra metric) using uniform exploration and dueling bandit algorithms. Similar to the model-free case discussed in section 6.4, the annotation complexity of uniform exploration grows as $O(k^2)$ but the annotation complexity only varies as $O(k)$ for RMED, RCS, and RUCB dueling bandit algorithms.