Active Evaluation: Efficient NLG Evaluation with Few Pairwise Comparisons

A Project Report

submitted by

M AKASH KUMAR

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
AND
MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

June 2021

THESIS CERTIFICATE

This is to certify that the thesis entitled **Active Evaluation: Efficient NLG Evaluation with Few Pairwise Comparisons**, submitted by **M Akash Kumar**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology** and **Master of Technology**, is a bonafide record of the research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Mitesh M. Khapra
Research Guide
Associate Professor
Computer Science and Engineering
IIT-Madras, 600 036

Prof. Kaushik Mitra
Research Co-Guide
Assistant Professor
Electrical Engineering
IIT-Madras, 600 036

Date: 18th June 2021

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my advisor, Dr. Mitesh Khapra, for the continued mentorship, guidance and support over the past four years. I also wish to thank my co-advisor, Dr. Kaushik Mitra, who gave me the freedom to work in my area of interest and have been supportive throughout the course of my Dual Degree project. I'm grateful to all my teachers, the Analytics Club, and the Computer Vision Intelligence Group at IIT Madras for providing me a solid foundation, which was essential for completing this work.

I've been fortunate to have worked with several researchers during my Dual Degree at IIT Madras. I wish to thank Dr. Balaraman Ravindran, Dr. Balaji Vasan Srinivasan, Preksha Nema, and Ananya Sai for their support and help. I'm immensely thankful to all my friends at IIT Madras for their help throughout the last five years. Finally, I would like to thank my parents and brother for all their support and encouragement, without which this would not have been possible.

ABSTRACT

KEYWORDS: Automatic Evaluation, Natural Language Generation, Online Learning, Dueling Bandits, Bayesian Deep Learning

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.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS **i**

ABSTRACT **ii**

LIST OF TABLES **vi**

LIST OF FIGURES **1**

1 INTRODUCTION **2**

1.1 Overview 2

1.2 Key Contributions 4

2 Active Evaluation Framework **5**

2.1 Problem Formulation and Setup 5

2.2 Identifying the top-ranked system 6

2.3 Choosing System Pairs for Comparison 7

2.3.1 Uniform Exploration 7

2.3.2 Dueling Bandit Algorithms 7

3 Pairwise Probability Models **11**

3.1 Pairwise Probability Models 11

3.2 Preprocessing Steps 12

4 Model-based Dueling Bandits **14**

4.1 Random Mixing 14

4.2 Uncertainty-aware Selection 14

4.3 UCB Elimination 16

5 Experimental Setup **18**

5.1 Tasks & Datasets . . . 18
5.1.1 Description . . . 18
5.1.2 Dataset Preprocessing . . . 19
5.2 Automatic NLG Evaluation Metrics . . . 20
5.2.1 Direct Assessment Metrics . . . 20
5.2.2 Pairwise Evaluation Metrics: . . . 23
5.3 Annotation Complexity Measure . . . 23
5.4 Hyperparameters Details . . . 24
5.4.1 Dueling Bandit Algorithms . . . 24
5.4.2 Model-based Algorithms . . . 25

6 Results & Discussion 27
6.1 Analysis of Dueling Bandit Algorithms . . . 27
6.1.1 Validity of Assumptions . . . 27
6.1.2 Annotation Complexity . . . 28
6.1.3 Top-rank Prediction Accuracy . . . 28
6.2 Performance of Evaluation Metrics . . . 30
6.3 Analysis of Model-based Algorithms . . . 33

7 Detailed Analysis 36
7.1 Effect of number of NLG systems . . . 36
7.2 Comparison between Evaluation Metrics . . . 37
7.3 Sensitivity to Hyperparameters . . . 39
7.4 Robustness to Delayed Feedback . . . 39

8 Practical Recommendations & Best Practices 42
8.1 Practical Recommendations . . . 42
8.2 Best Practices for Choosing Hyperparameters . . . 43

9 Related Work 44

10 Conclusion & Future work 45

LIST OF TABLES

2.1 Various assumptions made by dueling bandit algorithms in the literature 8

2.2 Summary of dueling bandits algorithms in the literature along with their theoretical assumptions and the target winner of the learner 9

5.1 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 19

5.2 Tuned Hyperparameters of Model-based algorithms when used with the Electra Metric 25

5.3 Tuned Hyperparameters of Model-based algorithms when used with the Bleurt Metric 25

6.1 Validity of various assumptions made by Dueling bandit algorithms in different NLG datasets 28

6.2 Annotation complexity of the top 7 best performing dueling bandit algorithms along with the uniform exploration algorithm on 7 WMT datasets 29

6.3 Annotation complexity of the top 7 best performing dueling bandit algorithms along with the uniform exploration algorithm on Grammarly (FCE and Wki), E2E NLG, ParaBank and OpenAI TL;DR datasets 29

6.4 Sentence-level accuracy of direct assessment metrics with three probability models and our trained Electra metric in predicting the comparison outcome on WMT, Grammarly and CoNLL'14 shared task 32

6.5 Sentence-level accuracy of direct assessment metrics with three probability models and our trained Electra metric in predicting the comparison outcome on E2E NLG, ParaBank and OpenAI TL;DR dataset 32

6.6 Annotation complexity of model-based algorithms when used with RMED and Bleurt/Electra metric. 34

6.7 Annotation complexity of model-based algorithms when used with RMED and Bleurt/Electra metric. 34

LIST OF FIGURES

2.1 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. . 6

6.1 Top-rank prediction accuracy v/s number of human annotations used on WMT 16 tur-eng dataset 30

6.2 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 31

6.3 Sentence-level prediction accuracy of direct assessment metrics with the Linear, BTL, and BTL-Logistic models averaged across the 7 WMT datasets 32

6.4 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 35

7.1 Annotation complexity of (model-free) uniform exploration and dueling bandit algorithms v/s the number of NLG systems on the ParaBank dataset 36

7.2 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 37

7.3 Annotation complexity of Random Mixing with RMED using BLEU, Emnedding Average, Laser, MoverScore, BertScore, Bleurt and Electra evaluation metrics and standard RMED (model-free) 38

7.4 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 . . . 38

7.5 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 40

7.6 Annotation complexity of UCB Elimination with Bleurt v/s the Copland threshold for $\alpha = 0.6$ 40

7.7 Annotation Complexity v/s delays in feedback on the WMT16 deu-eng dataset 40

CHAPTER 1

INTRODUCTION

1.1 Overview

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, 2020*a*,*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 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.

1.2 Key Contributions

1. We formulate the problem of finding the top-ranked NLG system in an Active Evaluation framework and empirically evaluate the performance of 13 dueling bandit algorithms in 13 NLG datasets spanning 5 tasks.

2. We propose three model-based dueling bandit algorithms to combine automatic evaluation metrics with human evaluations.

3. Through extensive experiments, we show that our proposed model-based dueling bandit algorithms reduces the number of human annotations by 89%.

4. Based on the results of our large-scale empirical study, we provide practical recommendations and best practices to efficiently identify the top-ranked system.

CHAPTER 2

Active Evaluation Framework

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

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 $Y_2^{(t)}$. The learner's objective is to find the top-ranked system with as few pairwise comparisons as possible. Note that the feedback $w^{(t)}$ indicates the preference on only one input sample and not the entire test dataset. We assume that the annotator preference is stationary over time and its distribution is denoted by $p_a(w|Y_1, Y_2)$. The dependence of p_a on the context and reference is omitted for brevity. The preference relation between the NLG systems is given by:

$$p_{ij} = E_{Y_1, Y_2 \sim \mathcal{D}_{ij}} E_{w \sim p_a(w|Y_1, Y_2)} w \tag{2.1}$$

where $D_{ij} = \{Y_1^{(i)}, Y_2^{(i)}\}_{i=1}^{m}$ is a dataset consisting of the generated outputs from the systems i and j respectively. Let $\Delta_{i,j} = p_{ij} - \frac{1}{2}$. We assume that the order of

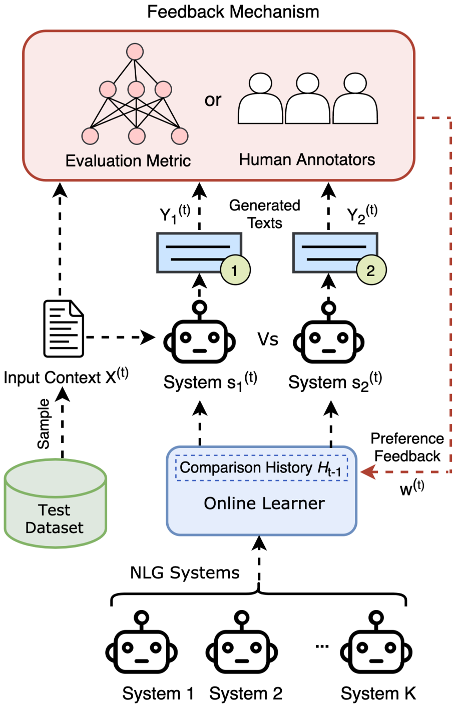

Figure 2.1: Our Active Evaluation framework consisting of a learner that chooses a pair of systems to compare at each time step. The learner receives feedback from either human annotators or the automatic metric.

displaying the systems does not affect the preference probabilities, hence $p_{ij} = 1 - p_{ji}$ *i.e.* $\Delta_{i,j} = -\Delta_{j,i}$. The overall framework is depicted in figure 2.1.

2.2 Identifying the top-ranked system

We now formalize the notion of the top-ranked system. We say that a system i beats system j if $\Delta_{i,j} = p_{ij} - \frac{1}{2} > 0$, *i.e.*, if the probability of winning in a pairwise comparison is larger for i than it is for j. A system i^* that beats all other systems, *i.e.* $\Delta_{i^*,j} > 0, \forall j \in \mathcal{S} - i^*$, is said to be a Condorcet winner. Note that a Condorcet winner need not always exist, and hence existing literature also considers the concept of a Copeland winner. A Copeland winner is the system that beats more systems than any other system does. However, in all our datasets and NLG tasks, we observed that the Condorcet winner exists. Therefore, we define the top-ranked NLG system as the Condorcet winner.

2.3 Choosing System Pairs for Comparison

2.3.1 Uniform Exploration

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

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

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 as we equally explore all $\binom{k}{2}$ pairs.

2.3.2 Dueling Bandit Algorithms

To reduce the number of annotations, we use dueling bandit algorithms that actively choose pairs of systems to compare based on the history of previous observations. Specifically, let $\mathcal{H}_{t-1} = \{s_1^{(\tau)}, s_2^{(\tau)}, w^{(\tau)}\}_{\tau=1}^{t-1}$ denote the observation history up to $t-1$, then the dueling bandit algorithm defines a mapping from $\mathcal{H}_{t-1}$ to system pairs $(s_1^{(t)}, s_2^{(t)})$. Many dueling bandit algorithms make assumptions on the true pairwise preferences and exploit these assumptions to derive theoretical guarantees (Bengs *et al.*, 2021). In table 2.1, 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 2.1, 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. We describe the 13 dueling bandit algorithms, that we analyze in this work, along with the assumptions and target winner in table 2.2. We provide an overview of these 13 algorithms below:

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 2.1: Various assumptions made by dueling bandit algorithms in the literature

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.*, 2017*b*), Sequential Elimination (Falahatgar *et al.*, 2017*a*), 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

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.*, 2017*a*)	SST	Condorcet
Plackett Luce (Szörényi *et al.*, 2015)	PL model	Condorcet
Knockout (Falahatgar *et al.*, 2017*b*)	SST+STI	Condorcet
Single Elim.(Mohajer *et al.*, 2017)	TO	Condorcet
RUCB (Zoghi *et al.*, 2014*b*)	CW	Condorcet
RCS (Zoghi *et al.*, 2014*a*)	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 2.2: Summary of dueling bandits algorithms in the literature along with their theoretical assumptions and the target winner of the learner

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.*, 2014*b*) 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.*, 2014*a*) 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.

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).

CHAPTER 3

Pairwise Probability Models

We discuss three pairwise probability models to convert the scores from direct assessment metrics into pairwise preference probabilities in 3.1. Later in 3.2, we discuss more implementation details.

3.1 Pairwise Probability Models

Our Active Evaluation framework, which we described in the previous chapter, 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 such as BLEU (Papineni *et al.*, 2002), BertScore (Zhang *et al.*, 2020), Bluert (Sellam *et al.*, 2020), *etc*, are designed for direct assessment and not directly suitable for pairwise evaluations. In this chapter, 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 3.2, 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.

3.2 Preprocessing Steps

We now discuss the preprocessing steps and the hyperparameters in the 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 3.1, 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.

CHAPTER 4

Model-based Dueling Bandits

In the previous chaper, 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

Inspired by algorithms in model-based reinforcement learning such as Dyna (Sutton, 1990), we mix the real and evaluation metric predicted feedback given to the learner. Specifically, given generated text Y_1 and Y_2 at 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 $1 - p_m$ *i.e.* the feedback given to the learner is:

$$\tilde{w}^{(t)} = \begin{cases} \hat{w}^{(t)}, & \text{w.p. } p_m \\ w^{(t)} \sim p_a(w|Y_1, Y_2), & \text{w.p. } 1 - p_m \end{cases}$$

where w.p. denotes "with probability", and p_m is the mixing probability hyperparameter that controls the ratio of estimated and real feedback. As with other hyperparameters, we choose $_m$ 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 $q_{\phi}(\theta)$ in a tractable family of distributions by minimizing the KL divergence between the candidate and true posterior. Gal and Ghahramani (2016) have shown that stochastic regularization techniques such as Dropout (Hinton *et al.*, 2012) can be used to perform approximate variational inference in neural networks. That is 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 Monte-carlo 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:

$$\mathbb{I}(w, \theta|Y_1, Y_2, \mathcal{D}^{tr}) := \mathbb{H}(w|Y_1, Y_2, \mathcal{D}^{tr}) - E_{\hat{p}(\theta|\mathcal{D}^{tr})}\mathbb{H}(w|Y_1, Y_2, \theta)$$

As shown in (Gal *et al.*, 2017), we can approximate the BALD measure using samples from the Dropout distribution. Specifically, 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. Then, the BALD measure can be approximated 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. We estimate the Copeland score by predicting the pairwise preference probability between any two systems i and j as follows:

$$\hat{C}_i = \frac{1}{k-1} \sum_{j \neq i} \mathbb{1}(\hat{p}_{ij} > \frac{1}{2})$$
$$\hat{p}_{ij} = \frac{1}{N} \sum_{Y_1, Y_2 \in \mathcal{D}_{ij}} \hat{p}(Y_1 \succ Y_2 | \theta)$$

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 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 $\hat{\sigma}_{ij}^2$ is the estimated variance given by:

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

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 $\hat{C}_i^u = \frac{1}{K-1} \sum_{j \neq i} \mathbb{1}(\hat{u}_{ij} > \frac{1}{2})$. 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.

CHAPTER 5

Experimental Setup

We describe the (i) NLG tasks and datasets used in our experiments, (ii) automatic evaluation metrics used in our model-based algorithms, (iii) annotation complexity measure used for comparing dueling bandit algorithms, and (iv) hyperparameter details in our dueling bandit and model-based algorithms.

5.1 Tasks & Datasets

5.1.1 Description

We use a total of 13 datasets spanning 5 tasks in our experiments which are summarized in table 5.1.

Machine Translation (MT): We use 7 MT datasets from the WMT shared translation tasks conducted in 2015 and 2016 (Bojar *et al.*, 2015, 2016). Specifically, we use the human evaluations collected from the fin→eng, rus→eng, deu→eng language pairs in 2015 and tur→eng, ron→eng, cze→eng, deu→eng language pairs in 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 in the Cambridge Learner Corpus First Certificate in English (FCE), and (ii) formal articles in Wikipedia (Wiki). We also use the dataset collected by (Napoles *et al.*, 2015*a*) from the CoNLL-2014 Shared Task (Ng *et al.*, 2014) which involves grammatical correction of short English text written by non-native speakers.

Data-to-Text Generation: We use the human evaluation data released from the E2E NLG Challenge (Dusek *et al.*, 2020) where the task is to generate natural language utterance from a dialogue act-based meaning representation.

Paraphrase Generation: For Paraphrase Generation, we use the ParaBank dataset

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 5.1: 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

(Hu *et al.*, 2019) consisting of English paraphrases generated using Back Translation with various lexical constraints.

Summarization: We use the human evaluations (Stiennon *et al.*, 2020) of GPT3-like models (Brown *et al.*, 2020) on the TL;DR dataset (Völske *et al.*, 2017) mined from reddit posts.

5.1.2 Dataset Preprocessing

We now discuss the dataset 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.

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 3.1. 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 with the implementation details below.

5.2.1 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 briefly summarize them below:

1. **Chrf**: Chrf compares character n-grams between the hypothesis and the reference sentences. The metric computes n-gram precision and recall at the character level for various values of n (up to $n = 6$) and then combines them using arithmetic average to obtain the overall precision and recall. The final score is computed by taking a weighted harmonic mean between the overall precision and recall.

2. **BLEU**: BLEU computes word-level n-gram overlap between the hypothesis and the reference. It uses a clipped n-gram precision score where the count of an n-gram is clipped by the maximum number of times it appears in any of the references. It also includes a brevity penalty term where short hypotheses are penalized.

3. **ROUGE-L**: ROUGE-L utilizes the longest common sub-sequence between the hypothesis and the reference. Specifically, it is defined as the F-score between a precision and a recall score, calculated using the longest common sub-sequence between the hypothesis and the reference.

4. **Embedding Average:** Embedding Average metric computes an embedding for a sentence by averaging the word embeddings of the words present in the sentence. It then calculates the cosine similarity between the hypothesis sentence embedding and the reference sentence embedding.

5. **Vector Extrema** Vector Extrema defines the embedding for a sentence as the dimension-wise absolute maximum of word embeddings. Similar to Embedding Average, the final score is obtained by computing the cosine similarity between the hypothesis sentence embedding and the reference sentence embedding.

6. **Greedy Matching** Greedy Matching finds the closest match word in the reference for each word in the hypothesis based on cosine similarity of the word embeddings. It then obtains an aggregate score by averaging the closest match cosine distance over all words in the hypothesis. To make the metric symmetric, it repeats the same process, but now with reversed direction *i.e.*

the reference and hypothesis are interchanged. The final score is defined as the average score between the two directions.

7. **Laser:** Laser uses the multilingual contextualized word embeddings from a pretrained BiLSTM. The BiLSTM is used to jointly pretrained to learn multilingual sentence representations of 93 languages.

8. **BertScore:** BertScore computes the cosine similarity of BERT representations for each pair of words form the hypothesis and the reference. It then uses a greedy matching approach to calculate precision and recall scores between the reference and the hypothesis. The final metric is defined as the F-score between precision and recall.

9. **MoverScore:** MoverScore uses the Earth Mover distance to define an optimal matching between the hypothesis and the reference using contextualized embeddings from BERT.

10. **Bleurt:** The Bleurt metric uses a pretrained BERT model, which is specifically trained for NLG evaluation in a self-supervised fashion using various perturbations of Wikipedia sentences such as masked-infilling with BERT, back-translation, and dropping words. The Bleurt model is further fine-tuned on WMT direct judgments data.

Implementation Details: We use the nlg-eval library[1] 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 [2]. 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 4.2, we apply Dropout to the Bleurt model during test time to estimate prediction uncertainty.

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

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

5.2.2 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. For the summarization task alone, we couldn't find any pairwise human judgment dataset sufficient for finetuning the Electra model. We discuss the finetuning datasets and finetuning details below:

Finetuning Dataset: For Machine Translation, we used human evaluations of WMT 2013 and 2014, consisting of a total of 650k examples. For Grammatical Error Correction, we curated a training dataset of 180k pairs of texts and human preference using data released by (Napoles *et al.*, 2015*b*) 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.

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.

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 annotation 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 $\delta_{acc} = 0.05$).

5.4 Hyperparameters Details

We discuss the details of the hyperparameters and the tuning procedure used for dueling bandit algorithm in 5.4.1 and our model-based algorithm in 5.4.2. In all three cases, we use the validation split of the finetuning datasets described in 5.2.2 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.

5.4.1 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.*, 2014*a*), we use α (exploratory constant) = 0.501. For RUCB (algorithm 1 of (Zoghi *et al.*, 2014*b*)), 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 Elimina-

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 5.2: 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 5.3: Tuned Hyperparameters of Model-based algorithms when used with the Bleurt Metric

tion, 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.*, 2017*b*), 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.

5.4.2 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 5.2 and 5.3, 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 4.2. We set $L = 20$, *i.e.* we independently apply dropout 20 times for each test predictions.

CHAPTER 6

Results & Discussion

We discuss the results of various dueling bandits algorithms in 6.1, performance of evaluation metrics in 6.2 and our model-based algorithms in 6.3.

6.1 Analysis of Dueling Bandit Algorithms

We analyze (i) the validity of assumptions made by different dueling bandit algorithms, (ii) annotation complexity of dueling bandit algorithms in our 13 NLG datasets, and (iii) the top-rank prediction accuracy of dueling bandit algorithms for a given number of human annotaitons.

6.1.1 Validity of Assumptions

We now analyze the validity of the assumptions that we discussed in 2.3.2. In table 6.1, we report whether those assumptions hold in each of our 13 NLG evaluation datasets. We observe that Strong Stochastic Transitivity (SST) and Stochastic Triangle Inequality (STI) does not hold true in almost all the datasets. Further, the Total Order (TO) and Relaxed Strong Stochastic Transitivity (RST) does not hold true in a majority of the datasets. Of these assumptions, only the assumption on the existence of Condorcet winner (CW), hold true across all the datasets. We note here that we cannot verify the validity of these assumptions if we do not have access to the true human preference *i.e.* before performing human annotations, we cannot know if these assumptions hold true or not. We present these results here for the sake of analysis.

Task	Dataset	Assumptions				
		TO	CW	SST	RST	STI
Machine Translation	WMT15 fin→eng	True	True	False	True	False
	WMT15 rus→eng	False	True	False	False	False
	WMT15 deu→eng	False	True	False	False	False
	WMT16 tur→eng	True	True	False	True	False
	WMT16 ron→eng	True	True	True	True	False
	WMT16 cze→eng	True	True	False	True	False
	WMT16 deu→eng	False	True	False	False	False
Grammatical Error Correction	Grammarly (FCE)	True	True	False	True	False
	Grammarly (Wiki)	False	True	False	False	False
	CoNLL-2014 Shared Task	False	True	False	False	False
Data-to-Text	E2E NLG Challenge	False	True	False	False	False
Paraphrase	ParaBank	False	True	False	False	False
Summarization	TLDR OpenAI	False	True	False	False	False

Table 6.1: Validity of various assumptions made by Dueling bandit algorithms in different NLG datasets

6.1.2 Annotation Complexity

We report the annotation complexity of various dueling bandit algorithms on 7 WMT datasets in table 6.2 and the rest of the datasets in table 6.3. We observe that the annotation complexity of the uniform exploration algorithm is consistently high across all 13 datasets. In particular, the required human annotations become prohibitively expensive when the number of NLG systems is high. For example, in the E2E NLG (16 systems) and ParaBank dataset (28 systems), the annotation complexity is more than 65k and 820k, respectively. Further, algorithms like RUCB, RCS, RMED can exploit the CW assumption to quickly eliminate sub-optimal NLG systems and perform better than algorithms that don't make any assumptions (CCB, DTS, etc.). Third, RMED performs the best overall with an average reduction of 80.01% in human annotations compared with the uniform exploration algorithm.

6.1.3 Top-rank Prediction Accuracy

We now examine an alternative approach to assess the performance of dueling bandit algorithms. Annotation complexity measures the required number of human annotations to achieve 95% top-ranked prediction accuracy. We now fix the number of human annotations and compute the accuracy in predicting the

Algorithm	WMT 2016				WMT 2015		
	tur-eng	ron-eng	cze-eng	deu-eng	fin-eng	rus-eng	deu-eng
Uniform	19479	24647	10262	3032	2837	12265	17795
IF	117762	282142	135718	75014	101380	162536	261300
BTM	32010	17456	$> 10^5$	2249	2926	11108	8328
Seq-Elim.	10824	17514	5899	4440	16590	6881	17937
PL	7011	18513	4774	4618	7859	17049	15215
Knockout	3415	7889	4723	3444	5104	5809	5956
Single Elim.	4830	6000	5885	5340	6953	6465	6453
RUCB	3125	5697	3329	1636	**1655**	4536	6222
RCS	2442	**3924**	3370	1537	2662	3867	5296
RMED	**2028**	5113	**1612**	**864**	1707	**1929**	**4047**
SAVAGE	10289	18016	6639	2393	2675	12806	12115
CCB	7017	11267	5389	2884	4092	11548	10905
DTS	10089	9214	8618	4654	4850	13317	16473
DTS++	7626	9483	5532	2729	6465	9394	14926

Table 6.2: Annotation complexity of the top 7 best performing dueling bandit algorithms along with the uniform exploration algorithm on 7 WMT datasets

Algorithm			Grammarly		CoNLL	E2E	Para-	TL;
	rus-eng	deu-eng	FCE	Wiki	'14 Task	NLG	Bank	DR
Uniform	12265	17795	8115	34443	61369	65739	825211	5893
IF	162536	261300	226625	364304	713522	718492	605825	70071
BTM	11108	8328	2778	$> 10^6$	$> 10^6$	**2541**	10175	2038
Seq-Elim.	6881	17937	12851	48068	38554	41037	$> 10^6$	9046
PL	17049	15215	8037	13156	**5682**	60031	$> 10^6$	3871
Knockout	5809	5956	3134	3777	8055	7708	17418	4953
Single Elim.	6465	6453	6000	9000	12940	15000	55900	9045
RUCB	4536	6222	2732	5617	19024	10924	41149	1647
RCS	3867	5296	1816	**4606**	12678	7263	34709	1903
RMED1	**1929**	**4047**	**2093**	5647	9364	3753	**24132**	**1162**
SAVAGE	12806	12115	5767	22959	39208	41493	255208	4733
CCB	11548	10905	4386	10020	21392	16960	87138	2518
DTS	13317	16473	4355	11530	18199	19940	170467	1354
DTS++	9394	14926	9284	17774	31562	15065	52606	6284

Table 6.3: Annotation complexity of the top 7 best performing dueling bandit algorithms along with the uniform exploration algorithm on Grammarly (FCE and Wki), E2E NLG, ParaBank and OpenAI TL;DR datasets

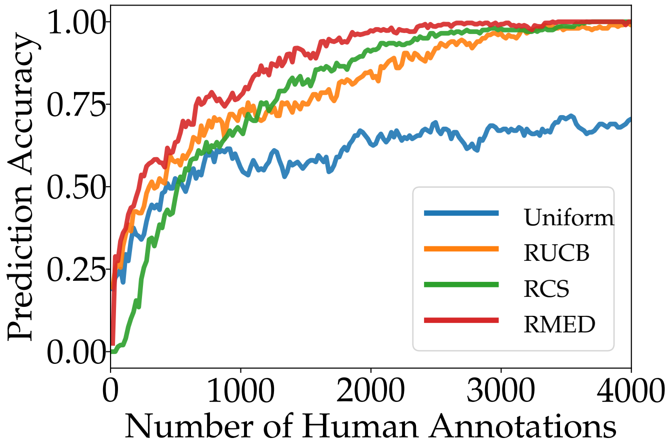

Figure 6.1: Top-rank prediction accuracy v/s number of human annotations used on WMT 16 tur-eng dataset

top-ranked system. That is we assume we have a fixed human annotation budget and we compute the top-ranked prediction accuracy obtained by the dueling bandit algorithms for the given number of annotations. In figure 6.1, we plot the top-ranked prediction accuracy as function of the number of human annotations for uniform exploration and the top three best performing dueling bandit algorithms, *viz.*, RUCB, RCS and RMED. We observe that RMED achieves the highest top-rank prediction accuracy for any given number of human annotations. As shown in figure 6.2, we observe similar trends for all 12 other datasets.

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. We report the sentence-level accuracy in predicting the comparison outcome w using existing direct assessment metrics with probability models (as discussed in 3.1) along with our trained pairwise evaluation model (Electra) in table 6.4 for the WMT, Grammarly, and CoNLL-2014 shared task datasets. For WMT and Grammarly, we report the Micro average accuracy across the 7 WMT and 2 Grammarly datasets. Similarly, in table 6.5, we report the sentence-level accuracy in predicting

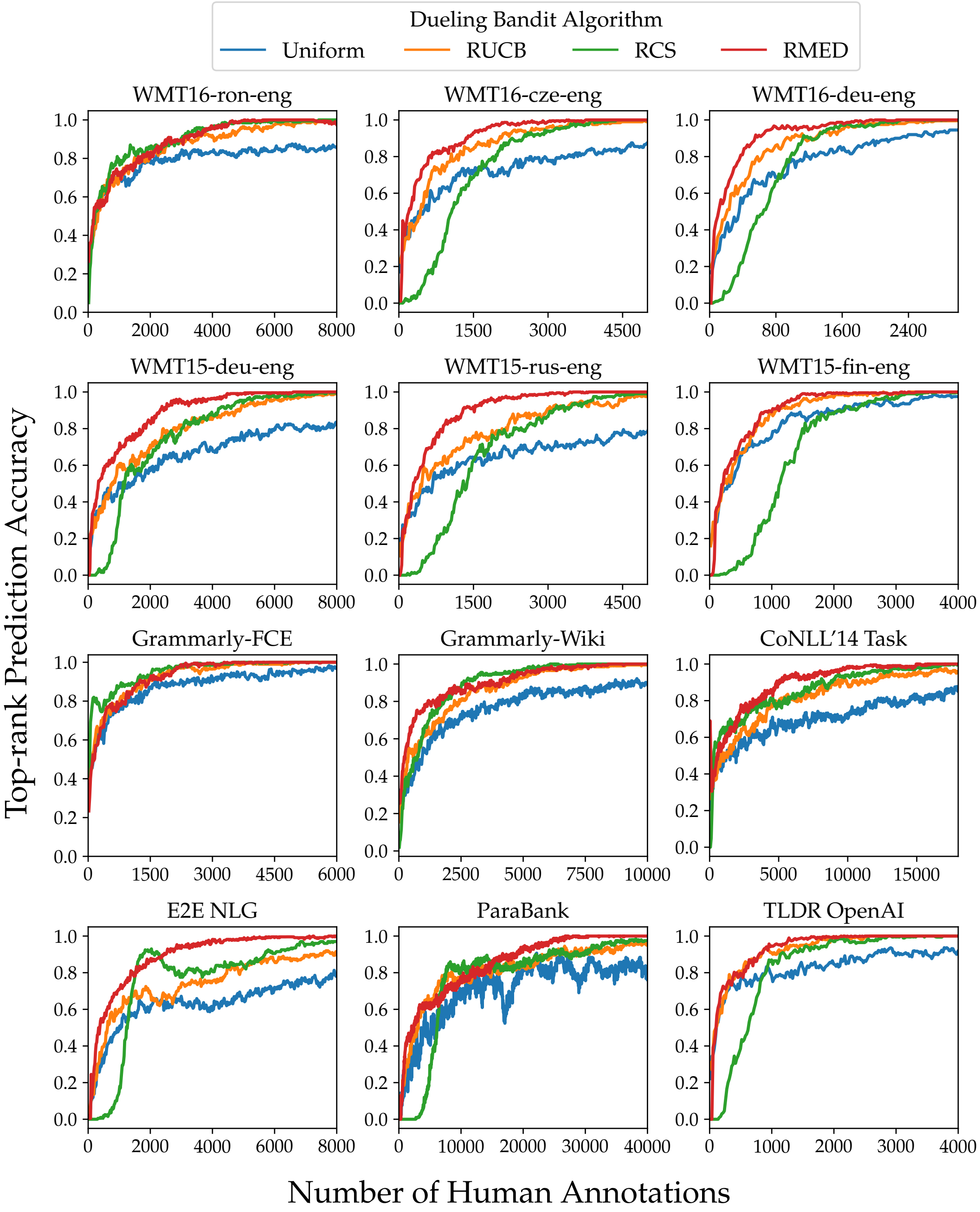

Figure 6.2: Top-rank prediction accuracy as a function of the number of human annotations for (model-free) Uniform exploration and RUCB, RCS, and RMED dueling bandit algorithms on 12 NLG datasets

Metrics	WMT (Micro Average)			Grammarly (Micro Average)			CoNLL-2014 Shared Task		
	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
Bleu-4	41.5	53.4	41.5	73.2	73.0	73.2	78.9	78.7	78.9
Rouge-L	60.7	60.0	60.7	73.5	73.6	73.6	78.0	78.0	78.0
Emb. Avg.	56.5	59.1	57.5	70.1	70.3	71.5	76.0	76.7	77.0
Greedy Match	59.5	59.8	59.9	68.1	68.4	68.2	77.7	77.4	77.7
Vector Extr	59.4	59.5	59.3	66.0	66.9	66.5	76.3	76.7	76.7
Bertscore	65.9	66.2	65.9	77.4	77.2	77.4	82.0	81.5	82.0
Laser	65.3	65.1	65.3	75.1	73.0	75.1	78.0	76.4	78.0
MoverScore	66.1	66.5	66.1	74.7	70.9	73.0	80.6	79.6	80.3
Bleurt	68.2	67.5	68.2	77.1	76.6	76.0	81.5	81.5	80.8
Electra	65.7			74.0			81.6		

Table 6.4: Sentence-level accuracy of direct assessment metrics with three probability models and our trained Electra metric in predicting the comparison outcome on WMT, Grammarly and CoNLL'14 shared task

Metrics	E2E NLG Challenge			ParaBank			TLDR OpenAI		
	Linear	BTL	BTL-log.	Linear	BTL	BTL-log.	Linear	BTL	BTL-log.
Chrf	47.4	48.8	48.3	66.1	66.1	66.1	34.2	35.4	35.4
Bleu-4	45.0	39.0	50.1	63.8	63.2	63.8	42.8	44.0	42.8
Rouge-L	44.6	43.8	50.2	64.3	64.3	64.3	43.3	43.3	43.3
Emb. Avg.	49.8	51.6	51.8	64.9	64.9	64.9	38.2	38.2	38.2
Greedy Match	46.5	48.8	48.9	64.7	64.7	64.5	43.1	43.1	43.1
Vector Extr	44.9	46.2	49.1	63.7	63.7	63.7	47.4	47.1	48.1
Bertscore	45.9	49.3	50.1	68.1	68.1	68.1	44.5	44.4	44.5
Laser	47.2	49.9	50.5	67.0	67.0	67.0	35.4	35.4	35.4
MoverScore	50.1	49.3	50.4	68.0	68.0	67.8	40.7	40.7	40.7
Bleurt	48.1	50.4	50.4	67.7	67.7	67.7	42.5	42.5	42.3
Electra	54.3			81.7			-		

Table 6.5: Sentence-level accuracy of direct assessment metrics with three probability models and our trained Electra metric in predicting the comparison outcome on E2E NLG, ParaBank and OpenAI TL;DR dataset

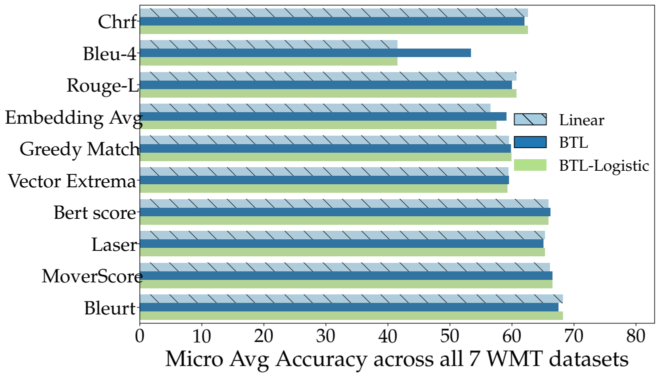

Figure 6.3: Sentence-level prediction accuracy of direct assessment metrics with the Linear, BTL, and BTL-Logistic models averaged across the 7 WMT datasets

the comparison outcome for E2E NLG Challenge, ParaBank and OpenAI TL;;DR datasets. We observe that *n*-gram and static word embedding-based metrics give a modest performance on the Machine Translation and Grammatical Error Correction tasks. When compared to these, we notice that metrics that utilize contextualized word embeddings, such as BertScore, perform much better. In Machine Translation, we observe that the Bleurt metric, which is specifically finetuned on WMT human judgment data, performs the best. On Data-to-Text generation and Paraphrase generation tasks, our trained Electra model 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 *i.e.* existing metrics do not add any useful value in evaluation. Hence, we exclude the TLDR dataset from our analysis on model-based dueling bandit algorithms. Lastly, 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 6.3 and observe that the performance is largely similar across Linear, BTL, and BTL-logistic models.

6.3 Analysis of Model-based Algorithms

We use our proposed model-based algorithms and incorporate the two best performing evaluation models, *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 6.6 for the 7 WMT datasets and in table 6.7 for the remaining datasets. Random mixing with Bleurt and Electra reduces the annotation complexity by 70.43% and 73.15% respectively on average 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 on average by 37.37% and 37.23% (when compared with random mixing) using Bleurt and Electra, respectively. We observe that the UCB Elimination algorithm also provide significant improvements in annotation complexity over the standard

Model-based	Evaluation	WMT 2016				WMT 2015		
Algorithm	Metric	tur-eng	ron-eng	cze-eng	deu-eng	fin-eng	rus-eng	deu-eng
None (Model free)	None	2028	5113	1612	864	1707	1929	4047
Random Mixing	Bleurt	237	1222	315	161	275	304	771
	Electra	728	3213	385	152	236	512	650
Uncertainty-aware	Bleurt	103	1012	192	84	204	239	530
Selection (STD)	Electra	978	7251	478	210	388	962	1259
Uncertainty-aware	Bleurt	101	653	**136**	48	181	162	405
Selection (BALD)	Electra	737	1648	223	114	207	538	488
UCB Eliminination	Bleurt	711	2684	1131	573	419	843	3556
	Electra	264	649	1131	414	294	1126	3556
Uncertainty	Bleurt	**31**	**415**	376	**25**	**59**	**82**	305
(BALD) + UCB Elim.	Electra	721	736	144	51	76	288	**280**

Table 6.6: Annotation complexity of model-based algorithms when used with RMED and Bleurt/Electra metric.

Model-based	Evaluation	Grammarly		CoNLL	E2E	Para-
Algorithm	Metric	FCE	Wiki	'14 Task	NLG	Bank
None (Model-free)	-	2093	5647	9364	3753	24132
Rand. Mix.	Bleurt	406	671	9584	1151	15874
	Electra	1529	237	3302	**326**	1044
Uncertainity (Mean STD)	Bleurt	270	185	9356	1291	22876
	Electra	477	234	4708	199	2137
Uncertainty (BALD)	Bleurt	**204**	**128**	**9356**	1167	22619
	Electra	281	75	1557	67	**858**
UCB Elimin.	Bleurt	967	1115	8382	2005	14098
	Electra	3970	1115	2943	1112	**9870**
Uncertainty (BALD) +	Bleurt	**162**	**39**	9995	256	4570
UCB Elim.	Electra	312	45	**782**	**40**	2247

Table 6.7: Annotation complexity of model-based algorithms when used with RMED and Bleurt/Electra metric.

RMED algorithm. Since UCB Elimination is complementary to Uncertainty-aware selection, we apply both these algorithms together and we observe the lowest annotation complexity with a reduction of 89.54% using Electra, and 84.00% using Bleurt when compared with the standard RMED algorithm.

In figure 6.4, 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.

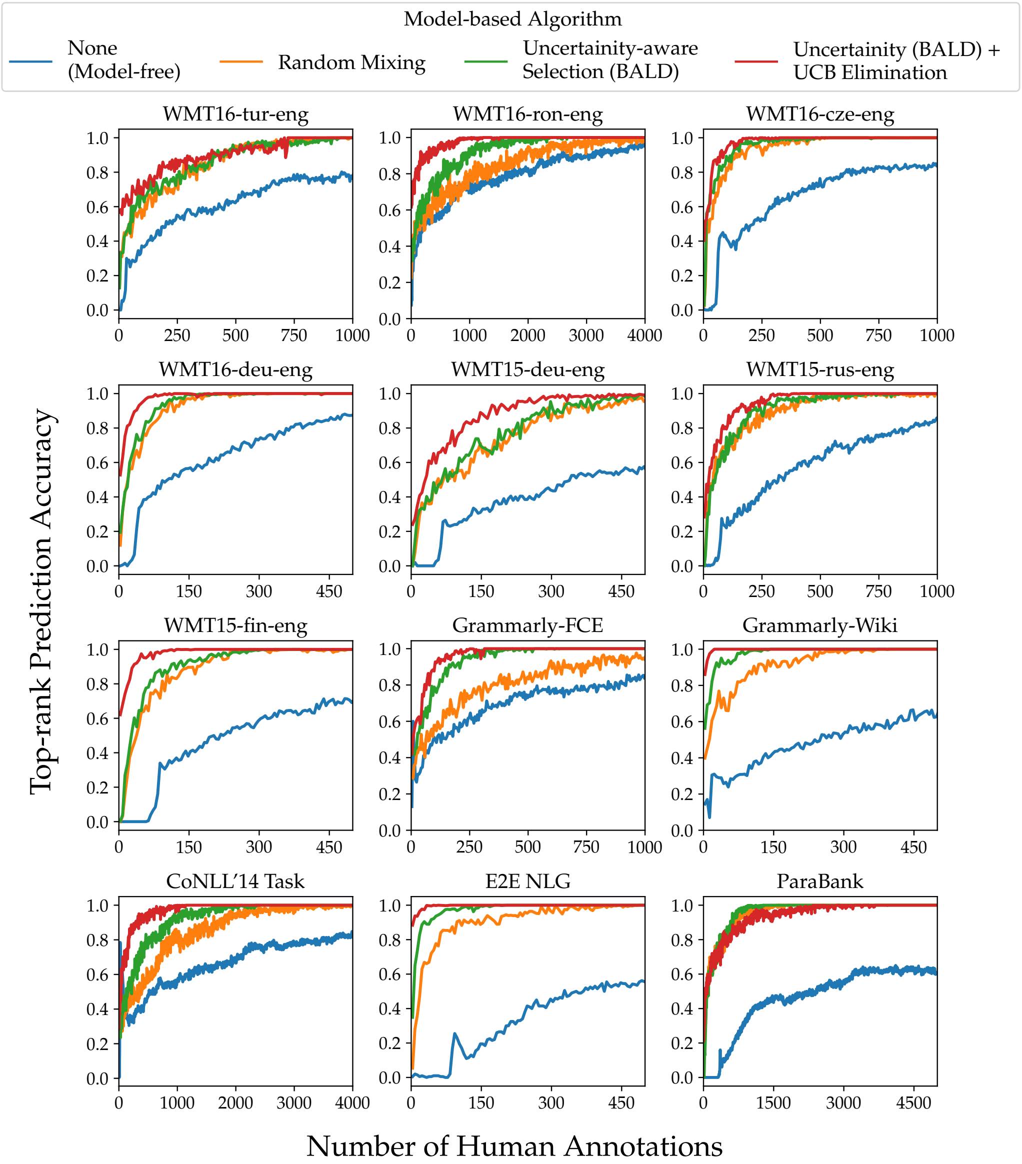

Figure 6.4: Top-rank prediction accuracy as a function of the number of human annotations for various model-based dueling bandit algorithms with RMED and Electra metric on 12 NLG datasets

CHAPTER 7

Detailed Analysis

We further analyze the factors affecting the query complexity of dueling bandit algorithms *viz.* number of NLG systems, automatic evaluation metrics, and hyperparameters of model-based algorithms. Lastly, we discuss the robustness of dueling bandit algorithms to delays in feedback arising when multiple human annotators are used in parallel.

7.1 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 7.1, 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 $O(k)$. Similarly, we compare the variations in annotation

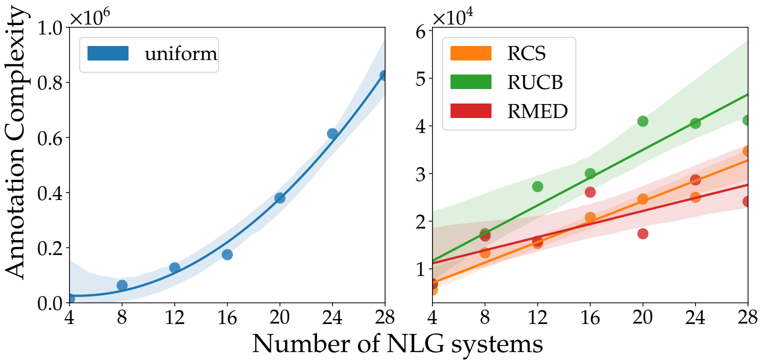

Figure 7.1: Annotation complexity of (model-free) uniform exploration and dueling bandit algorithms v/s the number of NLG systems on the ParaBank dataset

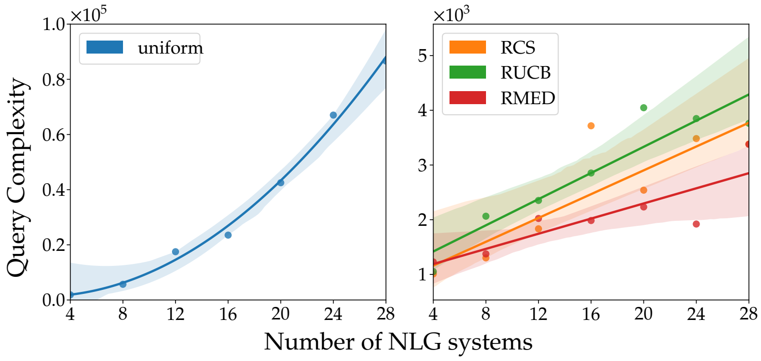

Figure 7.2: Annotation complexity of Random Mixing using the Electra metric with uniform exploration and dueling bandit algorithms as function of number of NLG systems on the ParaBank dataset

complexity when we use Random Mixing with the Electra metric. In figure 7.2, we plot the annotation complexity as function of number of NLG systems for uniform exploration and dueling bandit algorithms with Random Mixing. Like the model-free case, 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.

7.2 Comparison between Evaluation Metrics

We examine the effect of using different evaluation metrics in our random mixing algorithm with RMED in figure 7.3. We notice that using the BLEU metric, which only achieves an accuracy of 41.5% in MT, leads to an increase in query complexity in many Machine Translation datasets because of its inaccurate preference feedback. We observe similar trends for Embedding Average in a few datasets, but on average, we notice a decrease in query complexity by 7.95% relative to the model-free RMED algorithm. With Laser, MoverScore, and BertScore, we obtain an average reduction in query complexity by 51.27%, 57.34%, and 50.85%, respectively, relative to model-free RMED. The improvements are even greater with Bleurt and Electra at 70.43% and 73.15%, respectively.

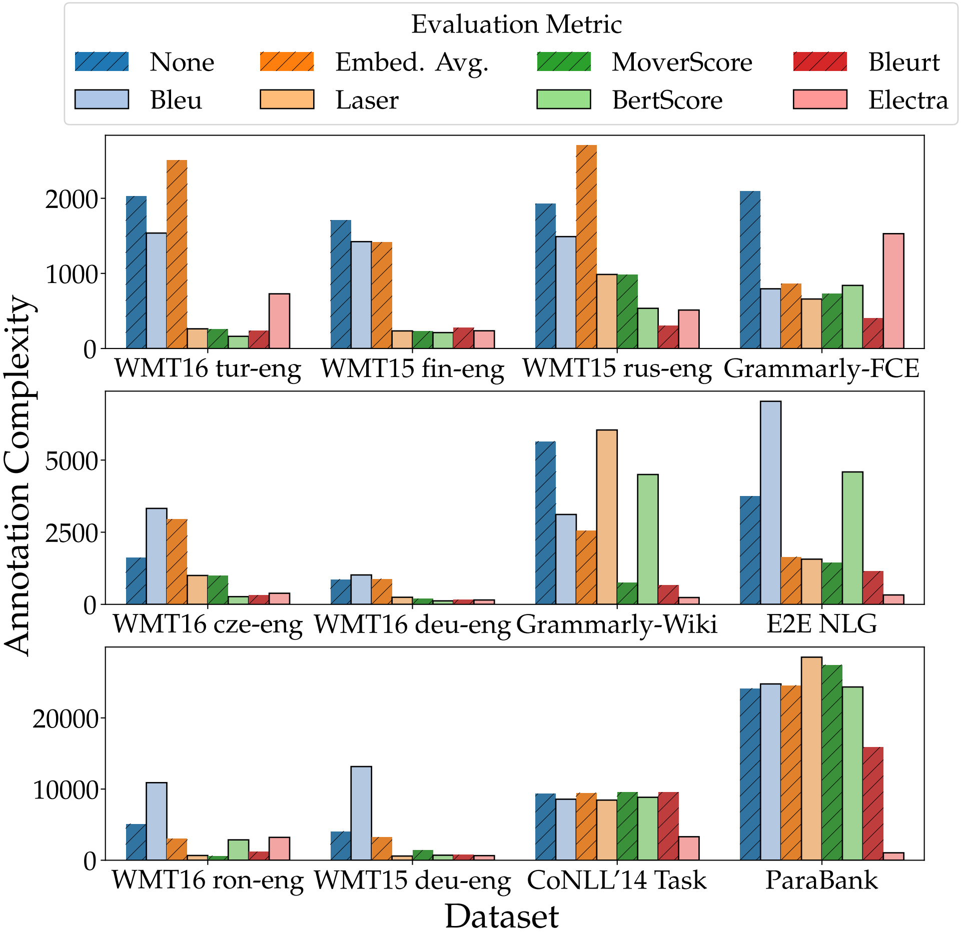

Figure 7.3: Annotation complexity of Random Mixing with RMED using BLEU, Emnedding Average, Laser, MoverScore, BertScore, Bleurt and Electra evaluation metrics and standard RMED (model-free)

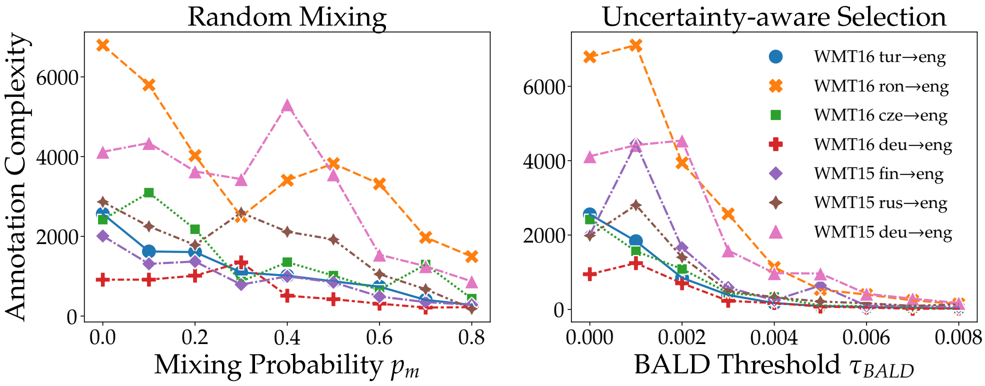

Figure 7.4: Variation in annotation complexity with Mixing probability in Random Mixing with Bleurt on the left and with BALD threshold in Uncertainty-aware Selection (BALD) with Bleurt on the right

7.3 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 7.4, 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 7.5, 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 7.6, 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.

7.4 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

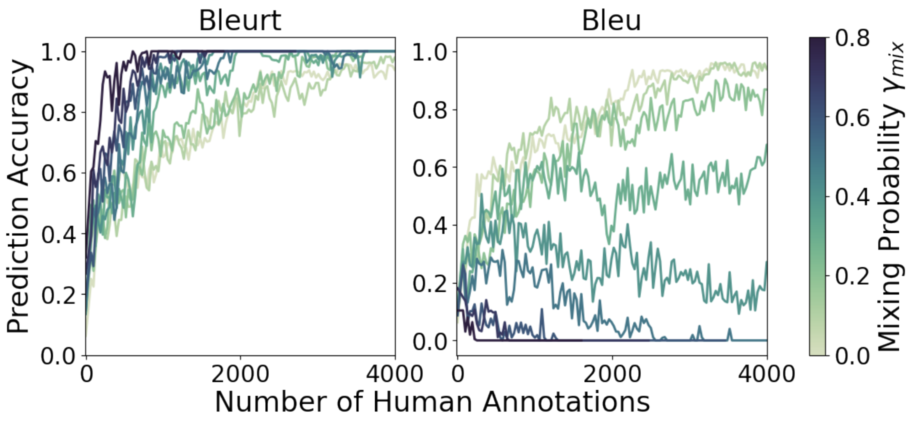

Figure 7.5: Prediction accuracy v/s number of human annotations collected for Random Mixing with Bluert and BLEU for different mixing probability p_m on the WMT 15 deu-eng dataset

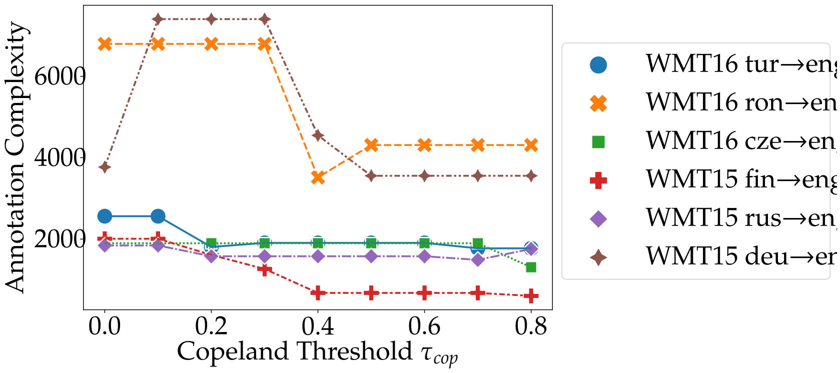

Figure 7.6: Annotation complexity of UCB Elimination with Bleurt v/s the Copland threshold for $\alpha = 0.6$

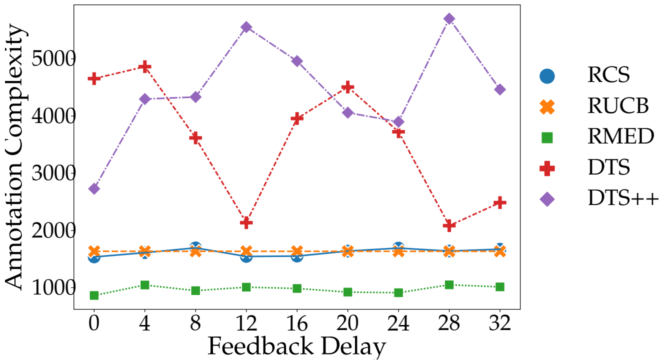

Figure 7.7: Annotation Complexity v/s delays in feedback on the WMT16 deu-eng dataset

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 7.7, 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.

CHAPTER 8

Practical Recommendations & Best Practices

8.1 Practical Recommendations

We summarize the key insights from this large-scale empirical 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 8.2 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.

8.2 Best Practices for Choosing Hyperparameters

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

CHAPTER 9

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. Apart from pairwise evaluations, a few other approaches (Novikova *et al.*, 2018; Kiritchenko and Mohammad, 2016) have been proposed in the literature to overcome the issues with direct assessment. Specifically, (Novikova *et al.*, 2018) proposes RankME, an extension of pairwise evaluation to multiple items. In RankME, human annotators rank several outputs provided to them. Another approach explored by (Kiritchenko and Mohammad, 2016) is Best–Worst Scaling where annotators are shown four outputs and asked to select the best and worst outputs. 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.*, 2014*b*,*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.

CHAPTER 10

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 where we actively decide the pairs of system to compare on one input sample from the test dataset. We used dueling bandit algorithms to choose the system pairs for comparison at each time instance. We extensively evaluated the performance of 13 dueling bandit algorithms proposed in the literature on 13 NLG evaluation datasets spanning five tasks. We showed that showed that dueling bandit algorithms can reduce the number of human annotations by 80% when compared to the uniform exploration baseline algorithm. We then proposed three model-based algorithms to combine automatic metrics with human evaluations. We showed that human annotations can be reduced further by 89% and thereby we required only a few hundred human annotations to identify the top-ranked system. We then provided practical recommendations and best practices to efficiently identify the top-ranked systems based on the results of our large-scale empirical study. In future work, we would like to extend our analysis to the general problem of finding the top-k ranked systems.

REFERENCES

1. **Artetxe, M.** and **H. Schwenk** (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Trans. Assoc. Comput. Linguistics*, **7**, 597–610. URL `https://transacl.org/ojs/index.php/tacl/article/view/1742`.

2. **Auer, P.**, **N. Cesa-Bianchi**, and **P. Fischer** (2002). Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, **47**(2-3), 235–256. URL `https://doi.org/10.1023/A:1013689704352`.

3. **Bahdanau, D.**, **K. Cho**, and **Y. Bengio**, Neural machine translation by jointly learning to align and translate. *In* **Y. Bengio** and **Y. LeCun** (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015. URL `http://arxiv.org/abs/1409.0473`.

4. **Belz, A.** and **E. Kow**, Discrete vs. continuous rating scales for language evaluation in NLP. *In The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*. The Association for Computer Linguistics, 2011. URL `https://www.aclweb.org/anthology/P11-2040/`.

5. **Bengs, V.**, **R. Busa-Fekete**, **A. E. Mesaoudi-Paul**, and **E. Hüllermeier** (2021). Preference-based online learning with dueling bandits: A survey. *J. Mach. Learn. Res.*, **22**, 7:1–7:108. URL `http://jmlr.org/papers/v22/18-546.html`.

6. **Bojar, O.**, **C. Buck**, **C. Federmann**, **B. Haddow**, **P. Koehn**, **J. Leveling**, **C. Monz**, **P. Pecina**, **M. Post**, **H. Saint-Amand**, **R. Soricut**, **L. Specia**, and **A. Tamchyna**, 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*. The Association for Computer Linguistics, 2014. URL `https://doi.org/10.3115/v1/w14-3302`.

7. **Bojar, O.**, **R. Chatterjee**, **C. Federmann**, **Y. Graham**, **B. Haddow**, **M. Huck**, **A. Jimeno-Yepes**, **P. Koehn**, **V. Logacheva**, **C. Monz**, **M. Negri**, **A. Névéol**, **M. L. Neves**, **M. Popel**, **M. Post**, **R. Rubino**, **C. Scarton**, **L. Specia**, **M. Turchi**, **K. M. Verspoor**, and **M. Zampieri**, 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*. The Association for Computer Linguistics, 2016. URL `https://doi.org/10.18653/v1/w16-2301`.

8. **Bojar, O.**, **R. Chatterjee**, **C. Federmann**, **B. Haddow**, **M. Huck**, **C. Hokamp**, **P. Koehn**, **V. Logacheva**, **C. Monz**, **M. Negri**, **M. Post**, **C. Scarton**, **L. Specia**, and **M. Turchi**, 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*. The Association for Computer Linguistics, 2015. URL `https://doi.org/10.18653/v1/w15-3001`.

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

10. **Brown, T. B.**, **B. Mann**, **N. Ryder**, **M. Subbiah**, **J. Kaplan**, **P. Dhariwal**, **A. Neelakantan**, **P. Shyam**, **G. Sastry**, **A. Askell**, **S. Agarwal**, **A. Herbert-Voss**, **G. Krueger**, **T. Henighan**, **R. Child**, **A. Ramesh**, **D. M. Ziegler**, **J. Wu**, **C. Winter**, **C. Hesse**, **M. Chen**, **E. Sigler**, **M. Litwin**, **S. Gray**, **B. Chess**, **J. Clark**, **C. Berner**, **S. McCandlish**, **A. Radford**, **I. Sutskever**, and **D. Amodei**, Language models are few-shot learners. *In* **H. Larochelle**, **M. Ranzato**, **R. Hadsell**, **M. Balcan**, and **H. Lin** (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html`.

11. **Clark, K.**, **M. Luong**, **Q. V. Le**, and **C. D. Manning**, 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, 2020. URL `https://openreview.net/forum?id=r1xMH1BtvB`.

12. **Dhingra, B.**, **M. Faruqui**, **A. P. Parikh**, **M. Chang**, **D. Das**, and **W. W. Cohen**, Handling divergent reference texts when evaluating table-to-text generation. *In* **A. Korhonen**, **D. R. Traum**, and **L. Màrquez** (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Association for Computational Linguistics, 2019. URL `https://doi.org/10.18653/v1/p19-1483`.

13. **Dusek, O.**, **J. Novikova**, and **V. 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. URL `https://doi.org/10.1016/j.csl.2019.06.009`.

14. **Elliott, D.** and **F. Keller**, 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*. The Association for Computer Linguistics, 2014. URL `https://doi.org/10.3115/v1/p14-2074`.

15. **Falahatgar, M.**, **Y. Hao**, **A. Orlitsky**, **V. Pichapati**, and **V. Ravindrakumar**, Maxing and ranking with few assumptions. *In* **I. Guyon**, **U. von Luxburg**, **S. Bengio**, **H. M. Wallach**, **R. Fergus**, **S. V. N. Vishwanathan**, and **R. Garnett** (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 2017*a*. URL `https://proceedings.neurips.cc/paper/2017/hash/db98dc0dbafde48e8f74c0de001d35e4-Abstract.html`.

16. **Falahatgar, M.**, **A. Orlitsky**, **V. Pichapati**, and **A. T. Suresh**, Maximum selection and ranking under noisy comparisons. *In* **D. Precup** and **Y. W. Teh** (eds.), *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*. PMLR, 2017*b*. URL `http://proceedings.mlr.press/v70/falahatgar17a.html`.

17. **Forgues, G.**, **J. Pineau**, **J.-M. Larchevêque**, and **R. Tremblay**, Bootstrapping dialog systems with word embeddings. *In NeurIPS, modern machine learning and natural language processing workshop*, volume 2. 2014.

18. **Gal, Y.** and **Z. Ghahramani** (2015). Bayesian convolutional neural networks with bernoulli approximate variational inference. *CoRR*, **abs/1506.02158**. URL `http://arxiv.org/abs/1506.02158`.

19. **Gal, Y.** and **Z. Ghahramani**, Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *In* **M. Balcan** and **K. Q. Weinberger** (eds.), *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*. JMLR.org, 2016. URL `http://proceedings.mlr.press/v48/gal16.html`.

20. **Gal, Y.**, **R. Islam**, and **Z. Ghahramani**, Deep bayesian active learning with image data. *In* **D. Precup** and **Y. W. Teh** (eds.), *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*. PMLR, 2017. URL `http://proceedings.mlr.press/v70/gal17a.html`.

21. **Herbrich, R.**, **T. Minka**, and **T. Graepel**, Trueskill$^{\text{tm}}$: A bayesian skill rating system. *In* **B. Schölkopf**, **J. C. Platt**, and **T. Hofmann** (eds.), *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*. MIT Press, 2006. URL `https://proceedings.neurips.cc/paper/2006/hash/f44ee263952e65b3610b8ba51229d1f9-Abstract.html`.

22. **Hinton, G. E.**, **N. Srivastava**, **A. Krizhevsky**, **I. Sutskever**, and **R. Salakhutdinov** (2012). Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, **abs/1207.0580**. URL `http://arxiv.org/abs/1207.0580`.

23. **Hitczenko, M.**, Modeling anchoring effects in sequential likert scale questions. 2013.

24. **Holtzman, A.**, **J. Buys**, **L. Du**, **M. Forbes**, and **Y. Choi**, The curious case of neural text degeneration. *In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=rygGQyrFvH`.

25. **Houlsby, N.**, **F. Huszar**, **Z. Ghahramani**, and **M. Lengyel** (2011). Bayesian active learning for classification and preference learning. *CoRR*, **abs/1112.5745**. URL `http://arxiv.org/abs/1112.5745`.

26. **Hu, J. E.**, **R. Rudinger**, **M. Post**, and **B. V. Durme**, 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*. AAAI Press, 2019. URL `https://doi.org/10.1609/aaai.v33i01.33016521`.

27. **Kendall, M.**, Rank correlation methods. 1948.

28. **Kiritchenko, S.** and **S. M. Mohammad**, Capturing reliable fine-grained sentiment associations by crowdsourcing and best-worst scaling. *In* **K. Knight**, **A. Nenkova**, and **O. Rambow** (eds.), *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*. The Association for Computational Linguistics, 2016. URL `https://doi.org/10.18653/v1/n16-1095`.

29. **Kitaev, N.**, **L. Kaiser**, and **A. Levskaya**, Reformer: The efficient transformer. *In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rkgNKkHtvB.

30. **Komiyama, J.**, **J. Honda**, **H. Kashima**, and **H. Nakagawa**, Regret lower bound and optimal algorithm in dueling bandit problem. *In* **P. Grünwald**, **E. Hazan**, and **S. Kale** (eds.), *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Workshop and Conference Proceedings*. JMLR.org, 2015. URL http://proceedings.mlr.press/v40/Komiyama15.html.

31. **Kulikov, I.**, **A. H. Miller**, **K. Cho**, and **J. Weston**, Importance of search and evaluation strategies in neural dialogue modeling. *In* **K. van Deemter**, **C. Lin**, and **H. Takamura** (eds.), *Proceedings of the 12th International Conference on Natural Language Generation, INLG 2019, Tokyo, Japan, October 29 - November 1, 2019*. Association for Computational Linguistics, 2019. URL https://aclweb.org/anthology/papers/W/W19/W19-8609/.

32. **Lewis, P. S. H.**, **E. Perez**, **A. Piktus**, **F. Petroni**, **V. Karpukhin**, **N. Goyal**, **H. Küttler**, **M. Lewis**, **W. Yih**, **T. Rocktäschel**, **S. Riedel**, and **D. Kiela**, Retrieval-augmented generation for knowledge-intensive NLP tasks. *In* **H. Larochelle**, **M. Ranzato**, **R. Hadsell**, **M. Balcan**, and **H. Lin** (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

33. **Li, M.**, **J. Weston**, and **S. Roller** (2019). ACUTE-EVAL: improved dialogue evaluation with optimized questions and multi-turn comparisons. *CoRR*, **abs/1909.03087**. URL http://arxiv.org/abs/1909.03087.

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

35. **Lin, C.-Y.**, ROUGE: A package for automatic evaluation of summaries. *In Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 2004. URL https://www.aclweb.org/anthology/W04-1013.

36. **Liu, Y.**, **J. Gu**, **N. Goyal**, **X. Li**, **S. Edunov**, **M. Ghazvininejad**, **M. Lewis**, and **L. Zettlemoyer** (2020). Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguistics*, **8**, 726–742. URL https://transacl.org/ojs/index.php/tacl/article/view/2107.

37. **Luce, R.**, Individual choice behavior: A theoretical analysis. 1979.

38. **Mathur, N.**, **T. Baldwin**, and **T. Cohn**, Sequence effects in crowdsourced annotations. *In* **M. Palmer**, **R. Hwa**, and **S. Riedel** (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. Association for Computational Linguistics, 2017. URL https://doi.org/10.18653/v1/d17-1306.

39. **Mohajer, S.**, **C. Suh**, and **A. M. Elmahdy**, Active learning for top-k rank aggregation from noisy comparisons. *In* **D. Precup** and **Y. W. Teh** (eds.), *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*. PMLR, 2017. URL http://proceedings.mlr.press/v70/mohajer17a.html.

40. **Moulin, H.**, Axioms of cooperative decision making. 1988.

41. **Napoles, C.**, **M. 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.

42. **Napoles, C.**, **K. Sakaguchi**, **M. Post**, and **J. Tetreault**, Ground truth for grammaticality correction metrics. *In ACL*. 2015*a*.

43. **Napoles, C.**, **K. Sakaguchi**, **M. Post**, and **J. Tetreault**, Ground truth for grammaticality correction metrics. *In ACL*. 2015*b*.

44. **Ng, H. T.**, **S. M. Wu**, **T. Briscoe**, **C. Hadiwinoto**, **R. H. Susanto**, and **C. Bryant**, The conll-2014 shared task on grammatical error correction. *In* **H. T. Ng**, **S. M. Wu**, **T. Briscoe**, **C. Hadiwinoto**, **R. H. Susanto**, and **C. Bryant** (eds.), *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2014, Baltimore, Maryland, USA, June 26-27, 2014*. ACL, 2014. URL `https://doi.org/10.3115/v1/w14-1701`.

45. **Novikova, J.**, **O. Dusek**, **A. C. Curry**, and **V. Rieser**, Why we need new evaluation metrics for NLG. *In* **M. Palmer**, **R. Hwa**, and **S. Riedel** (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. Association for Computational Linguistics, 2017. URL `https://doi.org/10.18653/v1/d17-1238`.

46. **Novikova, J.**, **O. Dusek**, and **V. Rieser**, Rankme: Reliable human ratings for natural language generation. *In* **M. A. Walker**, **H. Ji**, and **A. Stent** (eds.), *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)*. Association for Computational Linguistics, 2018. URL `https://doi.org/10.18653/v1/n18-2012`.

47. **Papineni, K.**, **S. Roukos**, **T. Ward**, and **W.-J. Zhu**, BLEU: a method for automatic evaluation of machine translation. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002. URL `https://www.aclweb.org/anthology/P02-1040`.

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

49. **Popovic, M.**, 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*. The Association for Computer Linguistics, 2015. URL `https://doi.org/10.18653/v1/w15-3049`.

50. **Radford, A.**, **J. Wu**, **R. Child**, **D. Luan**, **D. Amodei**, and **I. Sutskever**, Language models are unsupervised multitask learners. 2019.

51. **Radlinski, F.**, **M. Kurup**, and **T. Joachims**, How does clickthrough data reflect retrieval quality? *In* **J. G. Shanahan**, **S. Amer-Yahia**, **I. Manolescu**, **Y. Zhang**, **D. A. Evans**, **A. Kolcz**, **K. Choi**, and **A. Chowdhury** (eds.), *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*. ACM, 2008. URL `https://doi.org/10.1145/1458082.1458092`.

52. **Raffel, C.**, **N. Shazeer**, **A. Roberts**, **K. Lee**, **S. Narang**, **M. Matena**, **Y. Zhou**, **W. Li**, and **P. J. Liu** (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, **21**, 140:1–140:67.

53. **Ranzato, M.**, **S. Chopra**, **M. Auli**, and **W. Zaremba**, Sequence level training with recurrent neural networks. *In* **Y. Bengio** and **Y. LeCun** (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. 2016. URL `http://arxiv.org/abs/1511.06732`.

54. **Rennie, S. J.**, **E. Marcheret**, **Y. Mroueh**, **J. Ross**, and **V. Goel** (2017). Self-critical sequence training for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1179–1195.

55. **Roller, S.**, **E. Dinan**, **N. Goyal**, **D. Ju**, **M. Williamson**, **Y. Liu**, **J. Xu**, **M. Ott**, **K. Shuster**, **E. M. Smith**, **Y. Boureau**, and **J. Weston** (2020). Recipes for building an open-domain chatbot. *CoRR*, **abs/2004.13637**. URL `https://arxiv.org/abs/2004.13637`.

56. **Rus, V.** and **M. C. Lintean**, A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. *In* **J. R. Tetreault**, **J. Burstein**, and **C. Leacock** (eds.), *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, BEA@NAACL-HLT 2012, June 7, 2012, Montréal, Canada*. The Association for Computer Linguistics, 2012. URL `https://www.aclweb.org/anthology/W12-2018/`.

57. **Sai, A. B.**, **M. D. Gupta**, **M. M. Khapra**, and **M. Srinivasan**, 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*. AAAI Press, 2019. URL `https://doi.org/10.1609/aaai.v33i01.33016220`.

58. **Sai, A. B.**, **A. K. Mohankumar**, **S. Arora**, and **M. M. Khapra** (2020*a*). Improving dialog evaluation with a multi-reference adversarial dataset and large scale pretraining. *Trans. Assoc. Comput. Linguistics*, **8**, 810–827. URL `https://transacl.org/ojs/index.php/tacl/article/view/2389`.

59. **Sai, A. B.**, **A. K. Mohankumar**, and **M. M. Khapra** (2020*b*). A survey of evaluation metrics used for NLG systems. *CoRR*, **abs/2008.12009**. URL `https://arxiv.org/abs/2008.12009`.

60. **Sakaguchi, K.** and **B. V. Durme**, Efficient online scalar annotation with bounded support. *In* **I. Gurevych** and **Y. Miyao** (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. Association for Computational Linguistics, 2018. URL `https://www.aclweb.org/anthology/P18-1020/`.

61. **Sakaguchi, K.**, **C. Napoles**, **M. Post**, and **J. R. Tetreault** (2016). Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Trans. Assoc. Comput. Linguistics*, **4**, 169–182. URL `https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/800`.

62. **Sakaguchi, K.**, **M. Post**, and **B. V. Durme**, 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*. The Association for Computer Linguistics, 2014. URL `https://doi.org/10.3115/v1/w14-3301`.

63. **Sedoc, J.**, **D. Ippolito**, **A. Kirubarajan**, **J. Thirani**, **L. Ungar**, and **C. Callison-Burch**, Chateval: A tool for chatbot evaluation. *In* **W. Ammar**, **A. Louis**, and **N. Mostafazadeh** (eds.), *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*. Association for Computational Linguistics, 2019. URL `https://doi.org/10.18653/v1/n19-4011`.

64. **See, A.**, **S. Roller**, **D. Kiela**, and **J. Weston**, What makes a good conversation? how controllable attributes affect human judgments. *In* **J. Burstein**, **C. Doran**, and **T. Solorio** (eds.), *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)*. Association for Computational Linguistics, 2019. URL `https://doi.org/10.18653/v1/n19-1170`.

65. **Sellam, T.**, **D. Das**, and **A. P. Parikh**, BLEURT: learning robust metrics for text generation. *In* **D. Jurafsky**, **J. Chai**, **N. Schluter**, and **J. R. Tetreault** (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 2020. URL `https://doi.org/10.18653/v1/2020.acl-main.704`.

66. **Simpson, E. D.** and **I. Gurevych** (2018). Finding convincing arguments using scalable bayesian preference learning. *Trans. Assoc. Comput. Linguistics*, **6**, 357–371. URL `https://transacl.org/ojs/index.php/tacl/article/view/1304`.

67. **Srivastava, N.**, **G. E. Hinton**, **A. Krizhevsky**, **I. Sutskever**, and **R. Salakhutdinov** (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, **15**(1), 1929–1958. URL `http://dl.acm.org/citation.cfm?id=2670313`.

68. **Stiennon, N.**, **L. Ouyang**, **J. Wu**, **D. Ziegler**, **R. J. Lowe**, **C. Voss**, **A. Radford**, **D. Amodei**, and **P. Christiano** (2020). Learning to summarize from human feedback. *ArXiv*, **abs/2009.01325**.

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

70. **Sutton, R. S.**, Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. *In* **B. W. Porter** and **R. J. Mooney** (eds.), *Machine Learning, Proceedings of the Seventh International Conference on Machine Learning, Austin, Texas, USA, June 21-23, 1990*. Morgan Kaufmann, 1990. URL `https://doi.org/10.1016/b978-1-55860-141-3.50030-4`.

71. **Szörényi, B.**, **R. Busa-Fekete**, **A. Paul**, and **E. Hüllermeier**, Online rank elicitation for plackett-luce: A dueling bandits approach. *In* **C. Cortes**, **N. D. Lawrence**, **D. D. Lee**, **M. Sugiyama**, and **R. Garnett** (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. 2015. URL `https://proceedings.neurips.cc/paper/2015/hash/7eacb532570ff6858afd2723755ff790-Abstract.html`.

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

73. **Vaswani, A.**, **N. Shazeer**, **N. Parmar**, **J. Uszkoreit**, **L. Jones**, **A. N. Gomez**, **L. Kaiser**, and **I. Polosukhin**, Attention is all you need. *In* **I. Guyon**, **U. von Luxburg**, **S. Bengio**, **H. M. Wallach**, **R. Fergus**, **S. V. N. Vishwanathan**, and **R. Garnett** (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html`.

74. **Venkatesh, A.**, **C. Khatri**, **A. Ram**, **F. Guo**, **R. Gabriel**, **A. Nagar**, **R. Prasad**, **M. Cheng**, **B. Hedayatnia**, **A. Metallinou**, **R. Goel**, **S. Yang**, and **A. Raju** (2018). On evaluating and comparing open domain dialog systems. *arXiv: Computation and Language*.

75. **Vijayakumar, A. K.**, **M. Cogswell**, **R. R. Selvaraju**, **Q. Sun**, **S. Lee**, **D. J. Crandall**, and **D. Batra** (2016). Diverse beam search: Decoding diverse solutions from neural sequence models. *ArXiv*, **abs/1610.02424**.

76. **Völske, M.**, **M. Potthast**, **S. Syed**, and **B. Stein**, Tl;dr: Mining reddit to learn automatic summarization. *In* **L. Wang**, **J. C. K. Cheung**, **G. Carenini**, and **F. Liu** (eds.), *Proceedings of the Workshop on New Frontiers in Summarization, NFiS@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*. Association for Computational Linguistics, 2017. URL `https://doi.org/10.18653/v1/w17-4508`.

77. **Wang, S.**, **B. Z. Li**, **M. Khabsa**, **H. Fang**, and **H. Ma** (2020). Linformer: Self-attention with linear complexity. *ArXiv*, **abs/2006.04768**.

78. **Wieting, J.**, **M. Bansal**, **K. Gimpel**, and **K. Livescu**, Towards universal paraphrastic sentence embeddings. *In* **Y. Bengio** and **Y. LeCun** (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. 2016. URL `http://arxiv.org/abs/1511.08198`.

79. **Wu, H.** and **X. Liu**, Double thompson sampling for dueling bandits. *In* **D. D. Lee**, **M. Sugiyama**, **U. von Luxburg**, **I. Guyon**, and **R. Garnett** (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. 2016. URL `https://proceedings.neurips.cc/paper/2016/hash/9de6d14fff9806d4bcd1ef555be766cd-Abstract.html`.

80. **Xue, L.**, **N. Constant**, **A. Roberts**, **M. Kale**, **R. Al-Rfou**, **A. Siddhant**, **A. Barua**, and **C. Raffel** (2020). mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, **abs/2010.11934**. URL `https://arxiv.org/abs/2010.11934`.

81. **Yan, Y.**, **W. Qi**, **Y. Gong**, **D. Liu**, **N. Duan**, **J. Chen**, **R. Zhang**, and **M. Zhou** (2020). Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *ArXiv*, **abs/2001.04063**.

82. **Yue, Y.**, **J. Broder**, **R. Kleinberg**, and **T. Joachims** (2012). The k-armed dueling bandits problem. *J. Comput. Syst. Sci.*, **78**(5), 1538–1556. URL `https://doi.org/10.1016/j.jcss.2011.12.028`.

83. **Yue, Y.** and **T. Joachims**, Beat the mean bandit. *In* **L. Getoor** and **T. Scheffer** (eds.), *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*. Omnipress, 2011. URL `https://icml.cc/2011/papers/200_icmlpaper.pdf`.

84. **Zemlyanskiy, Y.** and **F. Sha**, Aiming to know you better perhaps makes me a more engaging dialogue partner. *In* **A. Korhonen** and **I. Titov** (eds.), *Proceedings of the 22nd Conference on Computational Natural Language Learning, CoNLL 2018, Brussels, Belgium, October 31 - November 1, 2018*. Association for Computational Linguistics, 2018. URL `https://doi.org/10.18653/v1/k18-1053`.

85. **Zhang, S.**, **E. Dinan**, **J. Urbanek**, **A. Szlam**, **D. Kiela**, and **J. Weston**, Personalizing dialogue agents: I have a dog, do you have pets too? *In* **I. Gurevych** and **Y. Miyao** (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. Association for Computational Linguistics, 2018. URL `https://www.aclweb.org/anthology/P18-1205/`.

86. **Zhang, T.**, **V. Kishore**, **F. Wu**, **K. Q. Weinberger**, and **Y. Artzi**, BERTScore: evaluating text generation with BERT. *In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=SkeHuCVFDr`.

87. **Zhao, W.**, **M. Peyrard**, **F. Liu**, **Y. Gao**, **C. M. Meyer**, and **S. Eger**, Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. *In* **K. Inui**, **J. Jiang**, **V. Ng**, and **X. Wan** (eds.), *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*. Association for Computational Linguistics, 2019. URL `https://doi.org/10.18653/v1/D19-1053`.

88. **Zoghi, M.**, **Z. S. Karnin**, **S. Whiteson**, and **M. de Rijke**, Copeland dueling bandits. *In* **C. Cortes**, **N. D. Lawrence**, **D. D. Lee**, **M. Sugiyama**, and **R. Garnett** (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. 2015. URL `https://proceedings.neurips.cc/paper/2015/hash/9872ed9fc22fc182d371c3e9ed316094-Abstract.html`.

89. **Zoghi, M.**, **S. Whiteson**, **M. de Rijke**, and **R. Munos**, Relative confidence sampling for efficient on-line ranker evaluation. *In* **B. Carterette**, **F. Diaz**, **C. Castillo**, and **D. Metzler** (eds.), *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*. ACM, 2014*a*. URL `https://doi.org/10.1145/2556195.2556256`.

90. **Zoghi, M.**, **S. Whiteson**, **R. Munos**, and **M. de Rijke**, 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*. JMLR.org, 2014*b*. URL `http://proceedings.mlr.press/v32/zoghi14.html`.