Master Thesis

Understanding Human Potentials for Evaluating Generative Models

Author(s):
Thorleiksdóttir, Thórhildur

Publication Date:
2021

Permanent Link:
https://doi.org/10.3929/ethz-b-000507443

Rights / License:
In Copyright - Non-Commercial Use Permitted
Master’s Thesis Nr. 348
Systems Group, Department of Computer Science, ETH Zurich

Understanding Human Potentials for Evaluating Generative Models

by
Thórhildur Thorleiksdóttir

Supervised by
Prof. Ce Zhang, Cedric Renggli, Dr. Nora Hollenstein

March 2021–August 2021
Abstract

Human evaluation is regarded as the primary metric when evaluating generative systems. In natural language generation, automatic metrics have reported flaws when applied to measure quality aspects of generated text and have been shown to correlate poorly with human judgments. However, human evaluations come with their own set of problems. They are expensive and time-consuming, and little consensus exists on conducting a human evaluation. Focusing on natural language generation, we propose a method to dynamically measure the required human annotations when evaluating models in a relative comparison setting. The approach ensures sufficient labelling to reach a confident decision on the optimal model with high probability when comparing two generative models. To examine the method across multiple labelling strategies, we implement an agent-based simulation framework of human evaluation for comparing two models simultaneously. Moreover, we conduct a human evaluation in a crowdsourcing setting to evaluate natural language generation models to examine the proposed decision method and labelling strategies. The main results indicate that we can make a high probabilistic decision on the better model when comparing two models with the proposed methodology. Moreover, by comparing multiple labelling strategies, assigning a single worker per request yields the least overall human effort to make a confident decision.
# Contents

Contents

## 1 Introduction

1

## 2 Background

2.1 Natural Language Generation

3

2.2 Evaluation of Natural Language Generation Systems

4

2.3 Human Labelling in Machine Learning

5

2.3.1 Human Annotated Training Data

5

2.3.2 Human Evaluation for Generative Models

7

2.4 Challenges in Human Evaluation

9

2.4.1 Data Collection Procedures

9

2.4.2 Lack of Consensus with Criteria

10

2.4.3 Recommendations for Human Evaluation

10

## 3 Methods

3.1 Two-Alternative Forced Choice Human Evaluation

13

3.2 Estimate Consistency for Highly Separable Models

14

3.2.1 Methodology Assumptions

15

3.2.2 Evaluation of Request Pairs

15

3.2.3 Estimate Consistency

16

3.3 Simulating Two-Choice Human Evaluation

17

3.3.1 Simulation Description

18

3.3.2 Estimate Decision Boundaries

21

3.3.3 Estimate Maximum Probability

22

## 4 Results

4.1 Agent-Based Human Evaluation

23

4.1.1 Experiment Setup

23

4.1.2 Distinguishable Models

24
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.3 Impact of Varying Human Capabilities</td>
<td>24</td>
</tr>
<tr>
<td>4.1.4 Decreasing Probability</td>
<td>31</td>
</tr>
<tr>
<td>4.1.5 Labelling Effort Distributions</td>
<td>33</td>
</tr>
<tr>
<td>4.1.6 Maximum Probability</td>
<td>33</td>
</tr>
<tr>
<td>4.1.7 Consistency for Distinguishable Models</td>
<td>36</td>
</tr>
<tr>
<td>4.1.8 Summary</td>
<td>36</td>
</tr>
<tr>
<td>4.2 Evaluation of Controlled Text Generation</td>
<td>40</td>
</tr>
<tr>
<td>4.2.1 Controlled Text Generation</td>
<td>40</td>
</tr>
<tr>
<td>4.2.2 Experiment Setup</td>
<td>40</td>
</tr>
<tr>
<td>4.2.3 Human Evaluation</td>
<td>43</td>
</tr>
<tr>
<td>4.2.4 Results</td>
<td>44</td>
</tr>
<tr>
<td>4.3 Evaluating End-to-End Text Generation</td>
<td>47</td>
</tr>
<tr>
<td>4.3.1 Two-Choice Evaluation for Data-Driven Text Generation</td>
<td>47</td>
</tr>
<tr>
<td>4.3.2 Experiment Setup</td>
<td>48</td>
</tr>
<tr>
<td>4.3.3 Results</td>
<td>50</td>
</tr>
<tr>
<td>5 Discussion</td>
<td>53</td>
</tr>
<tr>
<td>6 Conclusion</td>
<td>55</td>
</tr>
<tr>
<td>A Extended Simulation Analysis</td>
<td>57</td>
</tr>
<tr>
<td>A.1 Decreasing Probability</td>
<td>57</td>
</tr>
<tr>
<td>B Attribute Control Models</td>
<td>59</td>
</tr>
<tr>
<td>B.1 Automatic Evaluation</td>
<td>59</td>
</tr>
<tr>
<td>B.2 Generated Sentences</td>
<td>59</td>
</tr>
<tr>
<td>Bibliography</td>
<td>61</td>
</tr>
</tbody>
</table>
Natural Language Generation (NLG) systems are designed to generate text according to a defined task. Every day, people read or communicate with written text, and thus NLG systems main objective is to generate text in a clear and useful way. NLG is a subfield of Natural Language Processing (NLP), and text generation systems are often applied in various NLP applications such as language translation, chatbots, and summary creation, to name a few [12]. Evaluation remains challenging for NLG systems since text generation tasks are open-ended such that multiple correct responses exist. Moreover, we lack automatic metrics that successfully encode quality aspects of text. Therefore, human evaluation is regarded as the gold-standard evaluation metric for NLG systems [13, 12]. Still, evaluating systems with human judgements comes with several challenges. Human evaluations are expensive and time-consuming, especially when high-quality expert judgements are required or non-expert annotators need appropriate training. Moreover, due to budget limitations, evaluations are typically conducted with a fixed number of samples and annotators where many human evaluation studies are likely to be statistically underpowered to detect the true effects of evaluated models [11]. Comparing different systems solving the same tasks is difficult since there exists little consensus on how to design, conduct and report human evaluation [26]. The lack of agreement concerning evaluation design also impacts the consistency and quality of collected judgements [34, 41].

During the development of NLG systems, automatic metrics are generally applied to track progress even though common metrics such as BLEU [35] and ROUGE [29] have reported flaws and correlate poorly with human judgements [32, 21]. Human evaluation should have a bigger role during the development of NLG models to capture the progress of essential quality aspects required for generating text. Collecting human judgements regularly can support the development of automatic metrics to measure necessary quality aspects. Also, it enables standardising the experimental setup for
human evaluation, such as the task’s design, chosen data collection method, and labelling strategies.

In this thesis we aim to answer the following question: When comparing two natural language generation models, how can we conclude which system is better with a high probability according to a defined criteria without any pre-existing ground truth in a human evaluation setting? With this question in mind, we aim to relatively compare models with a two-alternative forced choice (two-choice) approach to collect sufficient information to make a confident decision. Performing an evaluation with relative comparisons can result in a higher inter-annotator agreement in contrast to evaluating models independently [10, 34] and its simplicity allows to design a straightforward evaluation task. We propose to use concentration inequalities to bound the model performance according to accumulated human judgements to make high probabilistic decisions on selecting the better model. As mentioned earlier, human evaluations are costly, and thus we focus on analysing the method using a simple agent-based human evaluation. We further aim to examine various labelling strategies to compare the required labelling effort to reach a confident decision with respect to the proposed decision method. With insights from our simulation experiments, we then analyse our approach by collecting actual human judgements when evaluating NLG systems.

In simulated and real human evaluation experiments, our results show that we can make a high probability decision (0.999) for all assessed labelling strategies when performing a two-choice human evaluation. Also, a comparison between annotation approaches yields that assigning a single worker per request requires the least labelling effort with respect to the proposed methodology.
Chapter 2

Background

2.1 Natural Language Generation

Natural language generation (NLG) systems, a sub-field of natural language processing (NLP), aim to automatically generate coherent and understandable text using non-linguistic information, single sentences or longer texts as input data [38, 12]. Everyday, people interact with applications where text generation is the core component of the system. Language translation for various languages, summarization of articles and documents, weather reports, and chatbots are examples of common NLG applications where the goal is to generate natural, fluent and grammatically correct text while remaining true to the system’s main communicative objectives. Different aspects of a language (e.g. linguistic structure, word usage, or grammar) need to be taken into account when building and training a language model, making it a challenging process. During the last decade, there has been a significant shift towards the usage of deep neural networks (DNN) as the state-of-the-art techniques for training NLG models. The paradigm shift started with recurrent neural networks (RNN) [46] applied with mechanisms such as long-short term memory (LSTM) [23] and gated recurrent units (GRUs) [14] for learning language representation such as word2vec, GloVe and sequence-to-sequence learning. Sequence-to-sequence models increased the usage of the encoder-decoder architecture in an unsupervised setting for various applications within the NLG domain [12] (e.g. variational autoencoder (VAE) [8]). Attention networks started to emerge due to the difficulty in capturing long-span dependencies in long text sequences with standard networks [2]. Instead of relying on sentences being compressed into a fixed-length vector (according to the sentence lengths in the training corpus), the model (soft-)searches relevant positions of the input sentences according to the recently predicted word. The model uses context vectors based on the searched positions to predict a target word instead of using a single fixed-length vector, allowing it to work better with longer sentences.
Recently, a model architecture called Transformers was proposed that relies entirely on the attention mechanism to align global dependencies between the input and output text of the network [49]. The Transformers architecture is currently being adopted as a new state-of-the-art neural language architecture within the NLG domain. Models based on the Transformers architecture (e.g. GPT-2 [37], GPT-3 [9]) trained on very large text corpora made up with online resources from Wikipedia articles, Reddit posts, search results, news articles, blogs, etc., can produce text that is nearly indistinguishable from human-generated texts [52, 15].

### 2.2 Evaluation of Natural Language Generation Systems

The continuous development of powerful language models emphasises the need for robust metrics for evaluating models from various perspectives. Different evaluation procedures can involve determining the diversity of the generated text, the quality aspects of the generated text (e.g. fluency, naturalness, or correctness), performing factual checks, or assessing how well the model fulfils its predefined task, to name a few. However, we can have various correct generated outputs according to the provided input like different responses in a chatbot setting or various summarisations of provided text document, making it challenging to evaluate NLG systems. Thus, most NLG tasks are open-ended which makes it difficult to apply automatic metrics to measure the performance of generated system, and thus human evaluation is considered as the gold standard for almost all NLG tasks. But it is expensive to conduct a human evaluation. Therefore, automatic metrics are generally applied to track daily progress for different aspects of the generated text for further model optimisations [12]. Several different metrics have been applied in the NLG domain over the years. Automatic metrics can be grouped into several categories: Metrics that are word overlap based metrics (e.g. BLEU, ROUGE, METEOR, F-SCORE), word embedding metrics (measure similarity using word embedding models such as Word2Vec) and focused automatic metrics. Focused automatic metrics can be applied to measure specific aspects of the generated text by using well-trained language models (often trained on human judgement data) to analyse the generated text in terms of fluency, correct style, diversity, or relevance to input [31]. Word overlap metrics or n-gram overlap metrics measure the similarity between the generated text and the human-generated ground-truth text. BLEU [35] was one of the first automatic metrics used to track the similarity between two sentences. It was designed initially for Machine Translation (MT) tasks but has also been applied in other NLG tasks. Earlier work reported that BLEU correlated well with human judgements in MT tasks. Still, more recent work has showed that it does not correlate well with human judgements for other NLG tasks, especially when tasks include
a lot of diversity for the generated text. It has been demonstrated that text with a perfect BLEU score can have poor information content and for unsupervised language generations, applying word overlap based metrics can potentially result in less accurate measurements [12]. Another word overlap based metric, ROUGE [29], was designed to evaluate multiple sentences or paragraphs, such as in summarisation tasks but is sometimes applied to evaluate short text generation (e.g. MT, image captioning and question generation). But ROUGE cannot give information about grammar, the narrative flow of the text and is unable to evaluate the factual correctness of the generated text in terms of its training data [12]. ROUGE also tends to favor longer generated text [21]. Current evolution procedures in various NLG tasks underlines existing challenges for evaluation purposes since measuring the progress of NLG models lacks standardised methods. As a result, authors select evaluation metrics with existing limitations [24, 32, 21], whether they are human-centric, automatic or machine-learned metrics.

### 2.3 Human Labelling in Machine Learning

In machine learning, humans have an essential role in providing necessary ground truth labels for training data and evaluating outcomes from models solving specific tasks. The correctness of the ground truth labels is vital since often, a model will only be as good as the provided data. Similarly, the correctness and consistency of human-labelled data for evaluation purposes is also important to understand the needed improvements of a given model. These two scenarios share similar challenges regarding reliability, consistency and accuracy regarding human performance and the design challenges of the data collection tasks. In the following sections, we will review related work regarding human-annotated training data to recognise methods relevant for data labelling strategies in human evaluation settings. Then we discuss current evaluation strategies for generated models, which include human judgements in their evaluation process.

#### 2.3.1 Human Annotated Training Data

In supervised learning, we need labelled training data where humans often produce the labelled datasets. Data is ever-growing, as well as the requirement of using more data to train models. Thus, human labelled data especially annotated by few selected experts becomes an expensive and time-consuming procedure. Therefore, a common practice is to hire non-experts annotators online through public crowdsourcing applications such as Ama-
2. **Background**

Mechanical Turk\(^1\), Appen\(^2\), or toloka.yandex\(^3\) to collect annotations for a given dataset [44, 51, 50, 45]. Multiple annotators annotate the same item to get a reliable consensus label representing the ground truth. A high level of agreement among annotators can further indicate that they share a similar understanding of the labelling task that can support the consistency of the produced annotations. Despite having consistency, the labels are not necessarily correct because we can have biased annotators agreeing on the wrong labels [36]. Label aggregation methods (e.g. majority voting) are applied to infer the ground-truth label using the collected annotations for each data item. Since the usage of crowdsourcing solutions has increased during the past decade to collect human-labelled data in various domains such as computer vision and NLP tasks [51, 45], there has been active research on improving label aggregation strategies when using crowdsourcing platforms. The usage of such platforms brings some theoretical and practical challenges. Labellers often have unknown expertise once they are hired to annotate data samples and there also exists the possibility of hiring adversarial labellers. Therefore, we have limited information about annotators before a task. Data items can also differ in their difficulty level, which can neither be known before the task. Due to these uncertainties, Whitehill et al. [51] proposed a probabilistic model of the labelling process, GLAD (Generative model of Labels, Abilities, and Difficulties). Standard inference methods were used simultaneously to infer the ability of the labellers, the difficulty of the data times, and the most probable label for each data item. Their results showed that GLAD outperformed the common majority voting method for aggregating labels both experimenting with simulated and real data, focusing on improving the accuracy of the labelling results.

When annotating data, it can be challenging to achieve high label accuracy when hiring annotators online. Annotators can be noisy without being adversarial. An intrinsically noisy annotator can be influenced, for example, by external factors (e.g. environment) or how clear the annotation guidelines are defined. Thus it is unlikely to achieve 100% accuracy for the aggregated labels through crowdsourcing. Another problem related to label creation for training data is the assumption that there only exists a single ground truth label for each data item, which goes against possible label ambiguities of data items. The denial of ambiguity [45] can, for example, generate arbitrary targets for machine learning models that focus on natural language understanding [17] since words can have several meanings. Recently, Sun and Kotel et al. [45] introduced a scalable methodology to estimate label accuracy generated by any crowdsourcing annotation methods without requiring any ground-truth information. Moreover, by perform-

\(^1\)https://www.mturk.com/
\(^2\)https://appen.com/solutions/crowd-management/
\(^3\)https://toloka.yandex.com
ing agent-based simulations to compare several annotation methods, they proposed DACR (Dynamic Automatic Conflict Resolution) to improve the annotation accuracy with less labelling efforts without neglecting potential labelling ambiguity.

2.3.2 Human Evaluation for Generative Models

Human evaluation is considered essential part for improving generative systems but results can be hard to reproduce and conducting evaluation with humans is expensive [12]. Several other issues exists regarding human evaluation that we will elaborate in more detail in Section 2.4. There have been attempts on making human evaluation less expensive such that it can become a more feasible evaluation approach. Chaganty and Mussman et al. [13] introduce human evaluation for NLG as an unbiased gold standard metric that is often too expensive. In order to achieve an unbiased estimate with lower cost than human evaluation alone, they propose a combination of automatic metrics with human evaluation using control variates but their results showed only 7 – 13% cost reduction compared to doing only human evaluations. They assume that improving bottlenecks regarding the evaluation design presented to humans and automatic metrics could result in greater cost-saving when combining statistical methods and human evaluation. The annotator variance and correlation between humans judgements and automatic metrics are the main properties that control the reduction in the amount of needed human annotations when combined with automated metrics to reach the same accuracy achieved with only conducting human evaluation.

Another drawback of conducting only human evaluation is that it fails to capture the diversity of the generative models. For example, humans are unable to state whether a model is plagiarizing sentences from its training data set and therefore it is hard to retrieve information regarding its ability to generate diverse data. To measure diversity, we can perform statistical evaluation by applying perplexity on a reference test. But perplexity has been shown being unable to measure quality [47] and other common metrics such as BLEU and ROUGE have showed bad correlation with human judgements [32]. As mentioned earlier, there exist approaches that combine statistical and human evaluation to measure both the diversity and quality of the generated text. According to Hashimoto and Zhang et al. [24], those approaches are ad-hoc and lead to misleading performance measures. Statistical measures provide an inadequate evaluation of quality aspects while human evaluators are unable to measure the diversity of a model. Instead, they propose an evaluation metric called HUSE, which combines human judgement and model probabilities to ensure a successful model performance both in terms of quality and diversity. The human judgements are collected
2. Background

to approximate the distribution for the reference sentences. That reference
distribution enables HUSE to estimate a two-dimensional output for each
generated sentence, representing the probabilities of a sentence coming from
the reference distribution or the model distribution. A simple KNN classi-
fier is used to classify which sentences are real or fake depending on these
two-dimensional probabilities.

Similarly to the goal for NLG models on generating realistic, grammatically
correct text, generating realistic images is one of the principal tasks when
measuring the progress of generative models in computer vision. Human
evaluations in computer vision tasks face similar problems as introduced in
human evaluation for NLG models. Methods are ad-hoc where estimates
result in a high variance since lacking details on task design can impact the
reproducibility of the results [40]. Therefore, it is hard to compare mod-
els and there does not exist a clear separability between evaluated models.
According to the law of large numbers, performing an experiment with a
large enough sample size of annotators and generated outputs should, the-
oretically, smooth out the variance and reach convergence but that is too
time-consuming and costly. With these limitations as motivations, Zhou
and Gordon et al. [53], proposed a gold standard human benchmark for
realistic image generation named HYPE which stands for Human eYe Per-
ceptual Evaluation. HYPE is grounded in psychophysics research in percep-
tual psychology and has the ability to separate the performance of models
with reliability, while also being efficient in cost and time. Out of two pre-
sented methods of HYPE, one has the goal of making the evaluation with
humans simpler, cheaper and faster. The HYPE score is computed as the
rate of mistakes when labelling real and fake images given unlimited time
to make decisions. HYPE is available for researchers where they can simply
upload their generative models using an online web platform to get a cor-
responding HYPE score which makes it a reasonable attempt to standardise
human evaluation for generated images. But in order to adapt HYPE for
NLG tasks, it would require a separate task’s design. The perceptual thresh-
old for images is much lower compared to reading and interpreting text [53]
which emphasises the challenges that remain for human evaluation for a
generated text.

Current goal in human evaluation is getting comparable results so that we
can have clear separability of models in terms of specified performance cri-
teria. Novikova et al. [34] created RankME, a human evaluation framework
designed to produce a reliable ranking of NLG models by combining con-
tinuous scales and relative assessments. Results show that it improves re-
liability and consistency compared to other standard evaluation methods
that use direct assessment. RankME is compared with these direct evalua-
tion methods in two different setups presented to the evaluators. The first
setup includes three criteria to be evaluated: Informativeness (also referred
2.4 Challenges in Human Evaluation

As mentioned already, human assessments are considered the gold standard evaluation metric for NLG but the literature demonstrates a little consensus on how such evaluations should be designed, conducted and reported. Howcroft et al. [26] analysed a dataset of 165 NLG papers that include details on human evaluations over the last two decades. The results show that there exist highly diverse approaches to perform human assessments on NLG systems. Different interpretations of various evaluation criteria and lack of reported experimental details necessary for reproducibility makes it difficult to compare results across various systems. Thus, there is an urgent need within the field of NLG for standardised methods and consistent terminology for human evaluation procedures. In this section we will discuss known limitations of human evaluation for text generation and discuss recent recommendations for collecting human judgements.

2.4.1 Data Collection Procedures

Before conducting a human evaluation, the first step is to select an appropriate data collection method, one of the key elements when designing an evaluation setup. Standard methods include collecting annotations using discrete scales such as rating scales (numerical and graphic scales) and Likert scales or applying continuous scales such as with magnitude estimation [43, 7]. These methods are applied to collect judgements for individual outputs (direct or absolute evaluation) or two or more outputs at a time (comparative or relative evaluation). An analysis of NLG evaluation methods showed that 63% of 135 observed papers applied rating scales or Likert scales. Likert scales have been controversial for years since researchers disagree whether they are considered ordinal scales or interval scales. Thus
one needs to be careful when applying parametric or nonparametric approaches to analyse collected results. Many papers in the study lack the justification for analysing the scales using associated statistics depending on whether they interpret the scales as ordinal or interval. That could indicate that authors are unaware of the current controversy regarding scale interpretations, leading to unwanted results such as overestimating or underestimating the differences between ratings [1]. Callison-Burch et al. [10] also noted that it is difficult for human judges to agree when judging data with Likert scales. Therefore judgements in this format can be inconsistent and not straightforward to compare. Comparative methods, where systems are evaluated against baseline models, variants of the implemented system or human-generated text, can lead to higher inter-annotator agreement [10]. Novikova et al. [34] reported that relative magnitude estimation for several systems solving the same task gives more distinct and consistent results in contrast to direct assessment methods with magnitude estimation or Likert scale. However, applying comparative methods can become costly, especially when comparing multiple systems that require many comparisons. Some methods try to reduce the annotation effort for multiple models, such as best-worst-scaling [30] to simplify relative evaluation [12]. Instead of ranking several models, participants select the best and worst model instead of choosing the corresponding rank for each candidate.

2.4.2 Lack of Consensus with Criteria

In 165 NLG papers that reported human evaluation results, more than 200 different terms describe quality aspects of generated text. Readability is an example of quality criteria, but it does not necessarily mean that it represents the same ‘readability’ aspects when applied for different systems. The same terms are often used to evaluate distinct quality aspects with their separate meaning, leading to confusion when comparing improvements for various systems [48, 26]. It is also a common issue that reported quality criteria have unclear or missing definition and lack corresponding evaluation setup (prompt/questions), making it difficult to interpret what is truly being evaluated [26]. More examples of quality criteria terms can include fluency, coherence, naturalness, quality, correctness, and informativeness. Sometimes, two metrics can be interpreted in the same way measuring the same thing (e.g. fluency and naturalness). Thus, it is crucial to report the meaning of the chosen criteria, its associated definition and designed setup such that the evaluation objectives are clear.

2.4.3 Recommendations for Human Evaluation

Human evaluation methods can be grouped according to intrinsic and extrinsic methods. The intrinsic evaluation focuses on specified criteria of
2.4. Challenges in Human Evaluation

the system’s output, such as asking evaluators to assess the naturalness or fluency of the generated outcomes. Extrinsic methods focus on evaluating how sufficiently the generative model achieves its main task. That is usually done by embedding the model to its target use context [48]. According to recent literature, intrinsic methods are more common since they tend to be less complex than extrinsic evaluation. The focus of improving NLG systems has also shifted towards improving subtasks of the NLG pipeline, which causes less need for extrinsic methods [48]. Due to these developments, aspects of text quality has become the primary evaluation measure and has been described with over 200 different terms over the last 20 years as mentioned above [26]. According to recent literature in human evaluation, van der Lee et al. [48] propose recommendations for the evaluation of an automatically generated text regarding the evaluation criteria, recruiting workers, and the evaluation design, to name a few. When conducting human evaluation, it is essential to use separate, properly defined criteria rather than assessing overall quality, which can be considered too abstract for annotators to measure directly [4]. Another good practice is to use a reader-focused design with a sufficient number of participants recruited rather than having a small number of experts. It has been found that experts approach evaluations differently such that their produced results can be affected by opinions and biases. That might not positively impact systems designed for the general population. Also, expert evaluations have shown limitations in predicting the outcome of reader-focused evaluation. For most NLG tasks, the number of recruited annotators should be three or more (two to three can be sufficient for a simple task) [48]. Fortunately, with crowdsourcing, it has become obtainable to recruit more workers for evaluation tasks, making it more feasible to hire the minimum number of workers for different evaluation tasks.
Chapter 3

Methods

3.1 Two-Alternative Forced Choice Human Evaluation

Two-alternative forced choice tasks [6] are designed such that a subject has to choose an answer from two available options, where the answer is selected for each task according to a given sensory input and a task description. A subject cannot choose a tie between two choices when not knowing the optimal response, and thus the process is subject to random fluctuations. The method enables retrieving the overall optimal choice with sufficient evidence for numerous tasks despite several random selections. This section introduces how an evaluation setup inspired by two-alternative forced choice is applied in our setting to evaluate generative systems simultaneously to conclude the optimal model. In the remainder of this thesis, we will refer to the two-alternative forced choice task as (forced) two-choice evaluation.

We want to test whether human evaluators, given two items at a given time, each sampled from different sets of items, can accumulate sufficient evidence over time to conclude which set is better according to pre-defined quality criteria where the sets are not equal in quality. Accordingly, given all those human annotations, can we reach a confident and consistent conclusion on deciding which set represents higher quality? We refer to quality as a generic term for various quality aspects of the provided items (e.g. for text we have quality aspects such as naturalness, fluency, readability, adequacy, correctness, etc.) [26].

The items in each set are outputs from generative models. An annotator selects the item with noticeable higher quality than the other item, without the knowledge of which models generated the items. If the annotator cannot distinguish the items, then a random choice is made by the annotator. The focus is not on optimising the accuracy of each individual pair being evaluated, but rather the overall performance of the accumulated information for multiple pairs of items. Thus loosing information regarding a pair being
indistinguishable is assumed to not have significant impact on the overall results. With majority of random selections for sufficient amount of given pairs an evaluation is expected to converge to a random decision, meaning that the two model candidates are not separable. One could argue that forcing the participants to choose between two options instead of additionally allowing them to label the items as indistinguishable or tied encourages the evaluators to aim for selecting the more suitable option according to their ability. If there is a negligible margin between the quality of the presented items, annotators preferences might impact the chosen answer in contrast to selecting a tie if the annotators are unsure, which might introduce bias (e.g. always preferring longer or shorter sentences).

By concluding which set holds higher quality for all request pairs enables us to state which model has relatively higher quality out of two generative models solving the same task. That is an essential insight into relative quality improvements when developing generative models especially since automatic metrics have been shown to be unable to measure necessary quality aspects of generated text as discussed in Chapter 2.

A two-choice evaluation gives an opportunity to design straightforward evaluation setup which enables fast evaluation procedure for annotators. Data is expected to be collected with less effort compared to performing separate direct assessments for each model. Also, evaluating models with relative comparison in previous work has resulted in a higher inter-annotator agreement compared to evaluating models independently [10, 34], which indicates the potentials for consistent evaluation among annotators.

In the following section, we attempt to adopt a theoretical framework proposed to study the latent accuracy for human classification of text. Applying the framework for human evaluation with two-choice approach enables us to estimate the consistency between independent evaluations of identical tasks when one model is assumed to perform better than the other on a pre-defined task.

3.2 Estimate Consistency for Highly Separable Models

Sun and Kotek et al. proposed a theoretical framework to analyse the latent accuracy for human-annotated text [45]. They derived a mathematical formula to approximate the proportion of requests that receive identical labels over independent annotation tasks for identical requests to estimate the expected performance of any annotation procedure. In their derivation, no pre-existing ground-truth is required but only the assumption that there exists a single correct answer\(^1\). In most text generation tasks, there exists

\(^1\)Assuming a single correct answer is a simplification for derivation purposes, but there can exist ambiguous answers in actual annotation tasks.
no ground truth for the generated data. Thus, analysing this framework in terms of human evaluation for two-choice evaluation can give valuable insights into estimating the consistency and the performance of the better model between two independent evaluations.

When models are easily distinguishable, a decision can potentially be made with less generated outputs and thus less cost compared to initialising a fixed sample size for evaluation purposes. Therefore it is important to estimate consistency and the performance of the better model, where fewer request pairs can be required to reach a decision.

3.2.1 Methodology Assumptions

Assume we have given two trained generative models \( A \) and \( A' \), developed to solve the same task, where we randomly sample \( n \) items from each of their corresponding latent spaces, \( a_i \sim z_A \) and \( a'_i \sim z_{A'} \), where \( 1 \leq i \leq n \). An annotation task consists of \( R \) requests such that \( n = |R| \), where each request \( r_i \) consists of one pair \((a_i, a'_i)\). To analyse the consistency for a two-choice procedure, we make the following assumptions:

- The annotators can only select one answer out of the two choices.
- One model has higher overall quality than the other model.
- The annotation procedure of each request is treated as an independent event and is not affected by prior events.
- We define a threshold \( \epsilon \), representing the lower bound of a noticeable difference in quality between the two items in each request pair. If the difference between the items is less than or equal to \( \epsilon \), then a choice is random.
- The noticeable difference \( \epsilon \) is strictly larger in each request pair, thus there exist a single correct choice for each pair\(^2\).

3.2.2 Evaluation of Request Pairs

We assume that model \( A \) is the better model in comparison to model \( A' \) in current setting. For all pairs \( \{(a_0, a'_0), (a_1, a'_1), \ldots, (a_n, a'_n)\} \) sampled from model \( A \) and \( A' \), we represent the difficulty of distinguishing items from model \( A \) with higher quality as a continuous random variable \( X \) over \([0, 1]\). The closer a random variable is towards 1 or 0 the easier it becomes to perceive items \( a \) or \( a' \) as the better item, respectively.

Therefore, \( X \) simply represents the likelihood \( P(l_r = a_r) \), of selecting item \( a \) with higher quality for a request pair \( r \) where \( l_r \in \{a_r, a'_r\} \).

\(^2\)In reality, we can expect more frequent cases where both items are indistinguishable according to given quality criteria.
3. Methods

Thus for any request \( r \), we make an independent random draw from \( X \) to assign the difficulty of selecting \( a \) of the corresponding pair. The expected mean of the request difficulties can be written as \( E[X] = \mu_X \) with variance \( \text{Var}[X] = \sigma_X^2 \).

As described in [45], we can treat the random variables for each request \( \{X_1, \ldots, X_n\} \) as a random sample of size \( n \), where the random variables are independent, identically distributed. Thus, we can then calculate the average of the random sample as:

\[
\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i
\]

Therefore, the sample mean \( \bar{X} \) is associated with the aggregated distribution of the proportion towards selecting \( A \) as the better model for all requests in \( R \).

Thus, according to the Central Limit Theorem [3], if we have a sufficiently large sample size \( n \) of random variables sampled from any distribution, the sample mean of the random sample will approximate to be associated with a normal distribution such that:

\[
\bar{X} \sim \mathcal{N}(\mu_X, \frac{\sigma_X^2}{n})
\]

with \( E[\bar{X}] = \mu_X \) and \( \text{Var}[\bar{X}] = \frac{\sigma_X^2}{n} \).

3.2.3 Estimate Consistency

As mentioned above, the likelihood of selecting item from model \( A \) as the correct choice is defined by the difficulty of distinguishing \( A \) from \( A' \) such that \( X_r \equiv P(l_r = a_r) \). Thus we denote the probability of selecting item \( A' \) as the choice with higher quality as \( P(l_r = a'_r) = 1 - P(l_r = a_r) \).

Now we want to compute the probability of receiving the same selection of items when performing the evaluation of the same pair twice. We set the answer for two evaluations \( l_r^{(i)} \) and \( l_r^{(j)} \) with likelihoods \( X_r^{(i)} \) and \( X_r^{(j)} \), where \( i, j \) represent different individual evaluations.

To compute the probability of getting the same selection for request \( r \), we can apply Bayes’ theorem such that:

\[
P(l_r^{(i)} = l_r^{(j)}) = P(l_r^{(i)} = l_r^{(j)} | l_r^{(i)} = a_r, l_r^{(j)} = a_r) \cdot P(l_r^{(i)} = a_r) \cdot P(l_r^{(j)} = a_r) + P(l_r^{(i)} = l_r^{(j)} | l_r^{(i)} = a', l_r^{(j)} = a') \cdot P(l_r^{(i)} = a') \cdot P(l_r^{(j)} = a')
\]

(3.3)
3.3 Simulating Two-Choice Human Evaluation

which can be simplified as:

\[ P(l_r^{(i)} = l_r^{(j)}) = 1 \cdot P(l_r^{(i)} = a_r) \cdot P(l_r^{(j)} = a_r) + 1 \cdot P(l_r^{(i)} = a') \cdot P(l_r^{(j)} = a') \]
\[ = P(l_r^{(i)} = a_r) \cdot P(l_r^{(j)} = a_r) + P(l_r^{(i)} = a') \cdot P(l_r^{(j)} = a') \]
\[ = X_r^{(i)} \cdot X_r^{(j)} + (1 - X_r^{(i)}) \cdot (1 - X_r^{(j)}) \]

(3.4)

With the given assumption that the items are easily distinguishable in terms of their quality, such that A is expected to have greater quality\(^3\), the task would need to fulfill:

\[ X_r^{(i)} \cdot X_r^{(j)} \gg (1 - X_r^{(i)}) \cdot (1 - X_r^{(j)}) \]

(3.5)

to approximate the likelihood of consistent labelling to the product distribution:

\[ P(l_r^{(i)} = l_r^{(j)}) = X_r^{(i)} \cdot X_r^{(j)} \]

(3.6)

If that requirement is met, the previous derivation enables us to express the consistency of evaluating identical requests as a continuous variable \( Y \) for request \( r \) for easily distinguishable items as the product distribution \( Y_r = X_r^{(i)} \cdot X_r^{(j)} \) with mean \( E[Y_r] = E[X_r^{(i)}] \cdot E[X_r^{(j)}] = \mu_X^2 \) and \( \text{Var}[Y] = (2\mu_X^2 + \sigma_X^2)\sigma_X^2 \) [22].

Thus we can estimate the consistency for a request pair between different annotations as \( E[Y] = \mu_X^2 \) when request pairs are easily distinguishable, such that \( A \) or \( A' \) is assumed being better. Thus, according to the given assumptions above, we can approximately derive the same results as presented in [45] to estimate the latent performance on how difficult it is to consistently separate two models in a two-choice evaluation. Evaluating the same requests in two separate evaluation procedures enables us to measure the ratio of requests that received identical labels as \( \hat{Y} \). Therefore we approximate the latent performance for selecting model \( A \) as \( \mu_X \approx \sqrt{\hat{Y}} \).

3.3 Simulating Two-Choice Human Evaluation

To examine the potentials of two-choice evaluation method for generative outputs, we introduce an agent-based simulation to analyse the performance and labelling effort of different labelling strategies. The required labelling effort represent the total number of labels we need to accumulate to make a confident decision. The simulation will further enable us to study the adapted framework to estimate the consistency when evaluating highly distinguishable models presented in section 3.2.

\(^3\)Identical computations can be applied when \( A' \) is assumed to have greater quality.
3. Methods

A human evaluation consists of multiple requests assigned to different human workers. Workers participating in an evaluation have varying capabilities, and requests differ in terms of difficulty (e.g. how difficult or easy it is to recognise the correct answer according to the goal of the task) [51, 11]. For various human evaluation experiments these parameters are rarely reported and collected evaluation data differ much based on the domain of the task, performance criteria, data collection methods (e.g. Likert scales or magnitude estimation), and number of collected evaluations [26, 11].

Previous work on comparing the accuracy between different data annotation methods using simulations apply probabilistic modelling of these parameters based on empirical observations [51, 45]. Thus we aim to similarly model the request difficulties and human capabilities in a human evaluation setting with probabilistic distributions. With the simulation we aim to examine the required labelling effort to make a confident decision with high probability for several labelling strategies according to pre-defined criteria. But first we need to answer how we can make a confident decision with collected judgements.

In the following sections, we will describe our simulation method where we further elaborate on the defined simulation parameters, different labelling strategies and how we abstract two-choice evaluation in the simulation. We then introduce our proposed decision method for two-choice human evaluation to conclude the better model with high probability.

3.3.1 Simulation Description

Similar to the formulation introduced in Section 3.2.1, we assume two generative models, $A$ and $A'$ designed to solve the same task. An evaluation consists of $n$ requests pairs $(a_i, a'_i)$ sampled from the latent spaces of the two generative models such that $a_i \sim z_A$ and $a'_i \sim z_{A'}$ where $1 \leq i \leq n$. Only one item in each request pair can be selected as the preferred item during an evaluation. The evaluation of a single request is an independent action and not affected by prior events.

Parameters

We initialise a uniform distribution, $c \sim Unif(a, b)$, where $a \geq 0$ and $b \leq 1$, which represent the annotators’ capabilities to distinguish two items for any request. Also, for the sake of simplicity, we exclude adversarial behaviour in our simulation framework. In a real crowdsourcing setting, there exist several methods that can be applied to reduce low-quality participation in annotation tasks [27, 50, 16].

For the requests difficulties $d$ we initialise a normal distribution $N(\mu, \sigma^2)$, where the mean $\mu$ varies between simulation experiments. The difficulty
levels are continuous random variables bounded between $[-1, 1]$. Thus we sample the difficulties as $d \sim \max(-1, \min(1, \mathcal{N}(\mu, \sigma^2)))$. The closer the mean is towards $-1$ or $1$, the easier it is to judge item from model $A'$ or $A$ as the better option, respectively.

For a coherent overview of the meaning of the lowest and highest values for human capabilities and request difficulties, we summarise important corner cases below:

- $c = 0$: Incapable annotator, not fluent in English and does not understand the task.
- $c = 1$: Highly capable annotator, fluent in English, strong grammatical skills, understands the task.
- $d = -1$: Easy to distinguish $a'$ as the better item compared to $a$.
- $d = 0$: Cannot distinguish $a$ being better than $a'$ (and vice versa).
- $d = 1$: Easy to distinguish $a$ as the better item compared to $a'$.

Formulation of the Evaluation Task

To simulate the evaluation of any request pair of items sampled from the latent spaces $z_A$ and $z_A'$ with any evaluator, we first compute the product $p = c \cdot d$. The product represents how difficult it is for a worker with capability $c$ to distinguish the better item of any request pair with difficulty $d$. We then transform the product from the range $[-1, 1]$ to a probabilistic range $[0, 1]$ to define the probability $P(a) = (p + 1)/2$ of selecting the item generated by model $A$ as the better item, and $P(a') = 1 - P(a)$ for choosing the item generated by $A'$. Finally, we abstract the evaluation of any request pair with a single Bernoulli trial with $P(1) = P(a)$ and $P(0) = P(a')$.

Table 3.1 shows the corner cases discussed above when computing the product $p = c \cdot d$ and the corresponding transformation of that result into a probabilistic variable. Evaluation with an incapable worker will result in random selection in all cases despite the task’s difficulty. A competent worker evaluates according to the associated difficulty. Thus if items are indistinguishable, the annotator choice will result in a random decision.

Labelling Strategies

As discussed in Chapter 2, there is no consensus on how human evaluations are conducted. For example, evaluation differs in the methods applied, the number of evaluated samples, how many workers are recruited and the design of evaluation tasks. Still, it is recommended to assign at least three workers per evaluation task [48]. Thus, we simulate the following labelling strategies:
3. Methods

Table 3.1: Computations associated with the meaning representation of the corner cases for human capabilities and request difficulties.

<table>
<thead>
<tr>
<th></th>
<th>$d = -1$</th>
<th>$d = 0$</th>
<th>$d = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c = 0$</td>
<td>$c \cdot d = 0$</td>
<td>$c \cdot d = 0$</td>
<td>$c \cdot d = 0$</td>
</tr>
<tr>
<td></td>
<td>$P(a) = P(a') = 0.5$</td>
<td>$P(a) = P(a') = 0.5$</td>
<td>$P(a) = P(a') = 0.5$</td>
</tr>
<tr>
<td>$c = 1$</td>
<td>$c \cdot d = -1$</td>
<td>$c \cdot d = 0$</td>
<td>$c \cdot d = 1$</td>
</tr>
<tr>
<td></td>
<td>$P(a) = 0, P(a') = 1$</td>
<td>$P(a) = P(a') = 0.5$</td>
<td>$P(a) = 1, P(a') = 0$</td>
</tr>
</tbody>
</table>

- **Fixed Worker**: The same worker is randomly selected to label all requests.
- **One Worker**: A different worker is randomly selected to label each request.
- **N Workers (Majority Vote)**: $N$ workers (randomly selected crowd-workers per request) where each worker labels given request. The majority will decide the final answer for a request where $N$ is an odd number.
- **Max 3 Workers**: Each request is randomly assigned to two workers. If they agree on the final answer, then that label is recorded. Otherwise, the request is assigned to one additional worker, which will determine the final answer.

The majority voting methods are often applied in human evaluation and data annotation tasks [51, 50, 28]. Assigning only a single worker per task is less common since it is generally recommended to hire several workers per task to measure the consistency between workers. Setting a maximum of three workers per task is inspired by Dynamic Automatic Conflict Resolution (DACR) and another former method called Double Grade, Conflict Resolved (DG, CR) [45]. The DG, CR relies on resolving annotation conflicts with experts, while the DACR strategy is implemented for multi-classification tasks for sentences to resolve conflicts dynamically without relying on expert annotations. The DACR method also returns high accuracy while reducing required labelling effort in comparison to majority voting methods.

**Measure Performance**

A final label per request is recorded depending on a given labelling strategy introduced above, where each label is selected according to pre-defined performance criteria. For each decided label, we update the proportion of selected choices with respect to the number of annotated request pairs. The recorded proportion measures the performance of the model comparison. In each simulation, we run multiple iterations with varying worker capabili-
ties for evaluating \( n \) identical requests. Therefore, we can estimate the mean proportion of accumulated choices over the increasing sample size for given request pairs.

### 3.3.2 Estimate Decision Boundaries

One of the main questions we want to answer is when can we decide with high probability which model is better with accumulated annotations according to a given labeling strategy? As mentioned before, we compute the proportion of the selections over the number of evaluated request pairs to measure the performance of the model comparison. Let \( X_i \) for \( 1 \leq i \leq n \) be a binary random variable such that \( X_1, \ldots, X_n \) represent final answer labels according to given labeling strategy for \( n \) requests. When \( X_i = 1 \), \( A \) is selected as the better item and when \( X_i = 0 \), \( A' \) is chosen as the better item. Thus,

\[
X = \frac{1}{n} \sum_{i=1}^{n} X_i
\]  

(3.7)

is the proportion of choices for selecting items generated by model \( A \). We assume that when the proportion of selections for a model is \( X > 0.5 \) that we can conclude the better model over \( n \) requests. But we want to be able to make such a decision with high probability.

*Concentration inequalities* are useful to bound the probability on how far a random variable deviates from the mean of its associated distribution. By applying a one sided version of Hoeffding inequality [25], we can bound the probability \( \delta \) with respect to the number of evaluated requests \( n \) and error tolerance \( t \) such that:

\[
\delta \leq e^{-2nt^2}
\]  

(3.8)

The bounded probability \( \delta \) represents the likelihood of the sample mean not being included within the given error bound:

\[
\delta = P(E[X] + t \leq X) = P(E[X] - t \geq X)
\]  

(3.9)

Accordingly, the probability of the sample mean being within the error bound is \( 1 - \delta \).

The results for a single human evaluation is represented with \( X \), and with multiple iterations we can further estimate \( E[X] \) in a simulating setting. But in practice, we lack information regarding \( E[X] \) when only conducting a single human evaluation. Thus we focus on computing the error bound with respect to the observed sample with sample mean \( \overline{X} \), with the same probability:

\[
\delta = P(\overline{X} - t \geq E[\overline{X}]) = P(\overline{X} + t \leq E[\overline{X}])
\]  

(3.10)
Thus, when $X > 0.5$ we make a decision when the corresponding lower bound satisfies $X - t > 0.5$ with $1 - \delta$ probability, where:

$$t = \sqrt{\frac{-ln(\delta)}{2n}}$$

(3.11)

is computed with fixed $\delta$ for increasing sample size $n$.

When we cannot reach a conclusion with sufficiently high probability, the models are indistinguishable according to the accumulated information.

### 3.3.3 Estimate Maximum Probability

Another strategy can be applied when modelling the decision process using Hoeffding inequality. Instead of pre-defining the target probability we can also find the maximum probability $1 - \delta$ for a given sample size $n$. We assume that a decision can be made when the lower bound is strictly larger than 0.5 or the upper bound is strictly lower than 0.5, and thus we can compute the intersection of the bounds with the decision boundary such that $t = 0.5 - X$ when $X < 0.5$ or $t = X - 0.5$ when $X > 0.5$. Thus with $t$ and sample size $n$ we can compute $\delta$ which results in the maximum probability $1 - \delta$ for a corresponding decision.

---

4Symmetric computations apply when $X < 0.5$ with respect to the upper bound.
Chapter 4

Results

4.1 Agent-Based Human Evaluation

With the introduced simulation approach in Section 3.3 we aim to answer if we can conclude the better model with high probability for all labelling strategies using the proposed simulation framework. We further examine if methods that rely on creating a final label per request with several workers (N Workers, Max 3 Workers) gain a performance advantage which enables decision making with fewer request pairs compared to assigning a single worker per request (Fixed Worker, One Worker).

4.1.1 Experiment Setup

We configure the distributions for the simulation parameters as shown in Table 4.1. The labelling of a request by a given worker depends on the computed probabilities, \( P(a) \) and \( P(a') \), as described in 3.3.1. To estimate the labelling performance, a simulation experiment consists of 1000 iterations where identical requests are evaluated for all labelling strategies with varying human capabilities in each iteration.

We perform simulated evaluations with three varying difficulty distributions such that all initialised distributions infer that model \( A \) is better than \( A' \), without the loss of generality, over all sampled requests. As mentioned earlier, we bound the request difficulty distribution to the range \([−1, 1]\) such that \( d \sim \max(−1, \min(1, N(\mu, \sigma^2))) \) when sampling from each of the initialised distribution in Table 4.1. The evaluated sample size depends on the request difficulty and worker capabilities and is configured such that a decision with sufficient probability is expected to be achieved in each iteration. Figure 4.1 shows the relative comparison of initialised request difficulty distributions for each simulation experiment.
4. Results

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Distribution</th>
<th>Sample Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workers ($c$)</td>
<td>$\text{Unif}(a,b)$</td>
<td>100</td>
</tr>
<tr>
<td>Request Diff. ($d^{(1)}$)</td>
<td>$\mathcal{N}(0.25,0.1)$</td>
<td>3500</td>
</tr>
<tr>
<td>Request Diff. ($d^{(2)}$)</td>
<td>$\mathcal{N}(0.125,0.1)$</td>
<td>8500</td>
</tr>
<tr>
<td>Request Diff. ($d^{(3)}$)</td>
<td>$\mathcal{N}(0.0625,0.1)$</td>
<td>30000</td>
</tr>
</tbody>
</table>

Table 4.1: Parameter configuration for the simulation experiments.

Figure 4.1: Probability density function of the sampled request difficulties for three simulation experiments.

4.1.2 Distinguishable Models

As discussed in Section 3.3.2, if $X > 0.5$ and the respective lower bound fulfills $X - t > 0.5$ a decision for the better model is made with $1 - \delta$ probability. Figure 4.2 shows the computed error bounds over increasing sample size for two separate simulations of the One Worker labelling strategy. For illustration purposes, we visualise the decision process with respect to the average proportion mean in defined simulation setting. The vertical dotted lines show the intersection of the computed lower bound with our defined decision threshold. The line shows that decision is achievable with $1 - \delta = 0.999$ probability in both simulations, but due to differences in request difficulties, the simulated evaluation with $\mu = 0.0625$ requires roughly $3\times$ more labelling effort than $\mu = 0.125$.

4.1.3 Impact of Varying Human Capabilities

In a non-expert crowdsourcing human evaluation, the capabilities of the workers play a significant role in the evaluation. Various aspects can have an impact on the abilities of workers. Usually, we have workers that perform
4.1. Agent-Based Human Evaluation

Figure 4.2: Computed error bound using Hoeffding inequality with respect to the proportional mean for selecting model A. Decision is made when the lower bound is strictly larger than 0.5, and the mean proportion is larger than 0.5. Human capabilities are sampled from $Unif(0.8, 1.0)$. 

(a) $\mu = 0.125$

(b) $\mu = 0.0625$
4. Results

a proper evaluation, where their capability differs on how well they understand the task with provided information as discussed in Section 3.3.1. In the following experiments, we analyse the impact of varying worker capabilities from having perfect annotators to workers that make more mistakes or are less qualified to conduct a defined evaluation. We expect that only the difficulty of given tasks affects the performance of ideal workers, but in practical crowdsourcing settings, workers have more diverse performance [50]. Thus we focus on examining if labelling strategies that require several workers per request gain performance advantage for varying worker capabilities and therefore require less labelling effort than assigning a single worker per request.

Ideal Annotators

To begin with, we want to analyse the ideal scenario where we have \( n \) requests, and all participating workers have perfect capability \( c = 1 \) where there is no variation between workers. How will diverse request difficulty impact the required labelling effort to reach a correct decision when all workers have ideal capability? According to our formulation of the request difficulties introduced in Section 3.3.1, any request pair with difficulty \( d = 0 \) represents indistinguishable items. Thus if we initialise a difficulty distribution with \( \mu = 0 \) in our simulation, we expect that the models cannot be separated as can be seen in Figure 4.3 or yield a random decision with increasing number of evaluated request pairs. In contrast, initialising a distribution with \( \mu = 0.25 \), we expect the results to yield \( A \) being the better model since the closer \( \mu \) is to 1, the easier it is to select \( A \) as the better model overall requests.

In the three simulation experiments for \( d^{(1)} \), \( d^{(2)} \), and \( d^{(3)} \) according to initialisation in Table 4.1 where \( a = b = 1 \), we expect that the simulated human evaluation performance converges such that \( A \) is the better model. But in each experiment, more request pairs will be required when it becomes harder to separate the models depending on how close \( \mu \) is to 0 and thus, labelling effort for each evaluation method is also expected to increase with more difficult evaluation.

Figure 4.4 shows four plots each representing a labelling strategy introduced in Section 3.3.1 for difficulty distribution with \( \mu = 0.125 \). Since \( c = 1 \) for all workers in the current experiment, One Worker has identical worker capability configuration as Fixed Worker, thus it yields the same results. All plots represent the mean proportion for selecting items generated from model \( A \) over 1000 iterations, computed over an increased number of request pairs. To recall, the dotted vertical line shows for how many requests the lower bound with respect to the proportion is strictly larger than 0.5 with \( 1 - \delta = 0.999 \) probability.
4.1. Agent-Based Human Evaluation

Figure 4.3: Mean proportion with associated decision bounds over an increasing number of request pairs with probability $1 - \delta = 0.999$ for 7 Workers and One worker where the difficulty distribution is initialised with $\mu = 0$. Human capabilities are sampled from $Unif(0.8, 1.0)$.

Table 4.2 summarises the average labelling effort for each set of difficulty level. As expected, increasing difficulty requires more labelling effort for all methods. Although 7 Workers and 5 workers with majority voting require on average fewer requests (as can be seen in Figure 4.4), they still require more labelling effort compared to other methods to reach a decision with the same probability. Thus for all difficulty levels, labelling all requests using a single worker with ideal capability requires the least labelling effort in contrast to the other examined labelling strategies.

Proper Workers

Experiments with our simulation method shows that when workers have ideal capability it is sufficient to assign a single worker per request to reach confident decision with the least labelling effort. Now we will analyse how more varied worker capabilities will impact the required labelling effort to reach a confident decision where the workers have proper capabilities. We run the same simulation experiment for identical request difficulty distributions as presented in Table 4.1 where $a = 0.8$ and $b = 1$, such that $c \sim Unif(0.8, 1.0)$.

Table 4.3 shows the average labelling effort for increased worker capability variance. Since we have more varied worker capability, each labelling strategy requires more labelling effort to reach a conclusion in comparison to results presented in Table 4.2 with ideal annotators. But when comparing the labelling efforts, it is interesting to note how similar the methods Fixed Worker and One worker perform in terms of required labelling effort.
In common crowdsourcing platforms, it is not common for the same worker to evaluate all requests since tasks are available for various workers at the same time when they are published. Crowdsourcing tasks can be designed to force the same worker to label multiple requests to get consistent labelling behaviour throughout the evaluation [53]. But forcing the same worker to label all requests sequentially is expected to be slower than hiring random workers per request, especially when a large sample size of requests is needed for evaluation. At the same time, we depend on the worker’s performance and with increasing requests, it can be more challenging for a worker to keep sufficient focus throughout the process. In contrast, hiring a different worker per request enables parallelization of the evaluation process, requiring less labelling time. Also, various workers assigned to each request can reduce the negative impact of getting one less capable worker for all requests.

The results in Table 4.3 show that we can make a decision with 0.999 prob-
### 4.1. Agent-Based Human Evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>Labelling Effort</th>
<th>99% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Workers</td>
<td>798 ± 252</td>
<td>778-819</td>
</tr>
<tr>
<td>5 Workers</td>
<td>634 ± 224</td>
<td>616-653</td>
</tr>
<tr>
<td>Max 3 Workers</td>
<td>389 ± 140</td>
<td>377-400</td>
</tr>
<tr>
<td>Fixed Worker</td>
<td>278 ± 119</td>
<td>267-287</td>
</tr>
<tr>
<td>One Worker</td>
<td>278 ± 119</td>
<td>267-287</td>
</tr>
</tbody>
</table>

\( \mu = 0.25 \)

<table>
<thead>
<tr>
<th>Method</th>
<th>Labelling Effort</th>
<th>99% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Workers</td>
<td>3199 ± 1410</td>
<td>3080-3316</td>
</tr>
<tr>
<td>5 Workers</td>
<td>2720 ± 1156</td>
<td>2635-2806</td>
</tr>
<tr>
<td>Max 3 Workers</td>
<td>1754 ± 707</td>
<td>1694-1819</td>
</tr>
<tr>
<td>Fixed Worker</td>
<td>1209 ± 469</td>
<td>1167-1245</td>
</tr>
<tr>
<td>One Worker</td>
<td>1209 ± 469</td>
<td>1167-1245</td>
</tr>
</tbody>
</table>

\( \mu = 0.125 \)

<table>
<thead>
<tr>
<th>Method</th>
<th>Labelling Effort</th>
<th>99% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Workers</td>
<td>12447 ± 3525</td>
<td>12166-12749</td>
</tr>
<tr>
<td>5 Workers</td>
<td>9739 ± 2928</td>
<td>9506-9980</td>
</tr>
<tr>
<td>Max 3 Workers</td>
<td>5920 ± 1898</td>
<td>5767-6070</td>
</tr>
<tr>
<td>Fixed Worker</td>
<td>3809 ± 1381</td>
<td>3700-3921</td>
</tr>
<tr>
<td>One Worker</td>
<td>3809 ± 1381</td>
<td>3700-3921</td>
</tr>
</tbody>
</table>

\( \mu = 0.0625 \)

Table 4.2: Labelling effort with standard deviation averaged over 1000 iterations for three difficulty distributions, where a decision is made with a probability of \( 1 - \delta = 0.999 \) and \( a = b = 1 \). The confidence intervals are computed with bootstrap resampling with 99% confidence.

ability with fewer workers evaluating more requests, requiring significantly less labelling effort compared to assigning multiple workers per request. According to computed confidence intervals, there is no statistical significance between the labelling effort required by Fixed Worker and One Worker. Nevertheless, One Worker allows for full parallelization and thus is a more viable option compared to the Fixed Worker.

**Increased Worker Variance**

Workers with capabilities sampled from the range \([0.8, 1.0]\) can be seen as proper annotators. But we want to push the variance of the capability range further apart to analyse the impact on the required labelling effort. Thus similar to the simulation procedure introduced above, we run the simulated evaluation according to the configurations from Table 4.1 and set \( a = 0.5 \) and \( b = 1 \) such that \( c \sim \text{Unif}(0.5, 1.0) \).

In Table 4.4 we can see that assigning fewer workers per request requires
4. Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Labelling Effort</th>
<th>99% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Workers</td>
<td>866 ± 306</td>
<td>841 – 888</td>
</tr>
<tr>
<td>5 Workers</td>
<td>722 ± 264</td>
<td>700 – 742</td>
</tr>
<tr>
<td>Max 3 Workers</td>
<td>461 ± 170</td>
<td>447 – 476</td>
</tr>
<tr>
<td>Fixed Worker</td>
<td>344 ± 168</td>
<td>331 – 357</td>
</tr>
<tr>
<td>One Worker</td>
<td>338 ± 154</td>
<td>325 – 349</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Labelling Effort</th>
<th>99% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Workers</td>
<td>3647 ± 1496</td>
<td>3536 – 3767</td>
</tr>
<tr>
<td>5 Workers</td>
<td>3141 ± 1256</td>
<td>3040 – 3236</td>
</tr>
<tr>
<td>Max 3 Workers</td>
<td>2011 ± 789</td>
<td>1952 – 2080</td>
</tr>
<tr>
<td>Fixed Worker</td>
<td>1454 ± 564</td>
<td>1404 – 1502</td>
</tr>
<tr>
<td>One Worker</td>
<td>1440 ± 553</td>
<td>1399 – 1489</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Labelling Effort</th>
<th>99% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Workers</td>
<td>13302 ± 3859</td>
<td>12965 – 13609</td>
</tr>
<tr>
<td>5 Workers</td>
<td>10850 ± 3271</td>
<td>10607 – 11114</td>
</tr>
<tr>
<td>Max 3 Workers</td>
<td>6729 ± 2172</td>
<td>6551 – 6900</td>
</tr>
<tr>
<td>Fixed Worker</td>
<td>4526 ± 1792</td>
<td>4376 – 4685</td>
</tr>
<tr>
<td>One Worker</td>
<td>4491 ± 1700</td>
<td>4356 – 4636</td>
</tr>
</tbody>
</table>

Table 4.3: Labelling effort with standard deviation averaged over 1000 iterations for three difficulty distributions and worker capabilities sampled from $Unif(0.8, 1.0)$, where a decision is made with a probability of $1 − \delta = 0.999$. The confidence intervals are computed with bootstrap resampling with 99% confidence.

significantly less labelling effort compared to assigning multiple workers per request. But with a more varied worker range, the required labelling effort increases further than having mostly proper workers across all methods for each distribution. Since we have less capable workers in this evaluation setting, we can also see the difference between Fixed Worker and One Worker in terms of the variance for needed labelling effort. The One Worker strategy has a smaller variance where it is less subject to hiring bad workers to evaluate all requests. But the standard deviation is surprisingly high across all previous experiments, which we will address further in Chapter 5. Similar to our previous results, despite more diverse worker capabilities, it is still optimal to assign a single worker per request to conclude which model is better with high probability where One Worker is the more viable option.
Table 4.4: Labelling effort with standard deviation averaged over 1000 iterations for three difficulty distributions and worker capabilities sampled from $\text{Unif}(0.5, 1.0)$, where a decision is made with a probability of $1 - \delta = 0.999$. The confidence intervals are computed with bootstrap resampling with 99% confidence.

### 4.1.4 Decreasing Probability

In previous section, we elaborated how varying human capabilities impact the required labelling effort when labelling identical request pairs. The same target probability $\delta = 0.001$ was applied when computing the error bounds for all previous experiments. Therefore, instead of having a fixed $\delta$, we want to understand how increasing $\delta$ affects the difference in labelling effort between given annotation strategies.

When $\delta = 1$, we have $1 - \delta = 0$ probability that the computed bounds will converge to the true distribution. The width of the error bound will be $t = 0$, and thus each method will only require a single request to reach a conclusion that will obviously be completely random since we have 0 confidence. To analyse the increasing $\delta \leq 1$, we run two experiments for the previously introduced difficulty distributions with $\mu = 0.25$, and $\mu = 0.125$ where we sample proper worker capabilities from $\text{Unif}(0.8, 1.0)$. For each $\delta$, we com-
Figure 4.5: Average labelling effort required for a decision over decreasing probability $1 - \delta$ with 99% confidence intervals.

Figure 4.5 represents the change in the required labelling effort on average when increasing $\delta$. For both experiments plotted in the Figures 4.5a and 4.5b results in a similar trend for increasing $\delta$. Assigning a single worker requires the least labelling effort over increasing $\delta$. But as discussed in earlier experiments, given such a similar labelling effort for Fixed Worker and One Worker, can further supports parallelism potentials of One Worker.

Figure 4.6 visualises the standard deviation (shaded area) of the recorded
labelling effort over 1000 iterations for each labelling strategy for \( \mu = 0.125 \), where \textit{Fixed Worker} and \textit{One Worker} yield the smallest variance over increasing \( \delta \). Also, requiring multiple workers per request indicates faster growth in terms of labelling effort when increasing the probability \( 1 - \delta \) (setting a smaller \( \delta \)) compared to assigning a single worker per request. Visualisation for \( \mu = 0.25 \) is presented in the Appendix in Figure A.1 and indicates similar trends.

4.1.5 Labelling Effort Distributions

In Section 4.1.3 we focused on analysing the impact of varying worker capabilities representing proper workers and its impact on the needed labelling effort. Extending the analysis for examining workers with \( a = 0.8 \) and \( b = 1 \), we further visualise the labelling effort distributions when making a decision with \( 1 - \delta = 0.999 \) probability over 1000 iterations for each difficulty distribution in Table 4.1.

Figure 4.7 shows the density curves for each labelling strategy for corresponding difficulty distribution. Across the distributions for each difficulty distribution, we can see several modes. Retrieved worker capabilities during the evaluation of identical requests control whether a labelling strategy can decide the better model with fewer requests early on or later. Often with fewer requests, especially in easier comparison scenarios (e.g. \( \mu = 0.25 \)), the performance is yet to converge. Thus, fluctuations in accumulated proportions over 1000 iterations can result in several modes for the observed labelling effort distributions depending on the sampled worker capabilities for each request in each iteration.

4.1.6 Maximum Probability

In Section 3.3.3 we discussed how the maximum probability of a decision being correct can be computed at the intersection with the upper or lower bound for a given sample size. To recall for the lower bound we find the size of the error bound as \( t = \bar{X} - 0.5 \) when \( \bar{X} > 0.5 \). Thus one might ask what the maximum probability is of a decision made with a given number of requests? Figure 4.8 shows the change in maximum probability when concluding a decision over increasing sample size for \textit{One Worker}. The plotted line shows the average probability over increasing sample size over 1000 iterations where the shaded area is the associated standard deviation for the computed probability. The vertical dotted line represents the intersection of the given sample size when probability \( 1 - \delta = 0.999 \) is achieved. We can see higher variance over smaller sample sizes for all difficulty distributions compared to the sample size represented by the dotted line. The variance of the computed probability becomes smaller and less fluctuating with a larger sample size since, with more samples, we become more confident about the
Figure 4.6: Average labelling effort with standard deviation over decreasing probability $1 - \delta$ with $\mu = 0.125$. 
Figure 4.7: Density curves for labelling effort distributions for each labelling strategy. Worker capabilities are sampled from $Unif(0.8, 1.0)$. 35
true distribution when bounding the evaluation performance with Hoeffding inequality. In the end, we can reach high probabilistic decisions, where the request difficulties and varying human capabilities control the required labelling effort.

4.1.7 Consistency for Distinguishable Models

Consistent reproducibility is important for a human evaluation, especially when performing evaluations with non-expert workers in a crowdsourcing setting. To estimate the performance by analysing the consistency between evaluation projects on the same request data, we introduced an adapted method to study the latent performance and consistency for a given labelling strategy for two-choice human evaluation in Section 3.2. We made a restricted assumption that the derivation only supports highly distinguishable models in the defined setting, where one model is easily evaluated as the better model. In this section, we will test the adopted method in our proposed simulation setting to analyse potential limitations regarding the required assumption for the defined framework.

Figure 4.9 shows the computed inter-project consistency rates and corresponding pairwise mean proportion for selecting model A as the better option when evaluating 10000 requests of varying difficulty for 100 iterations. The consistency rates are computed as the ratio of matching labels of evaluated requests between two iterations. The pairwise mean proportion is the mean between final aggregated proportions for the corresponding two iterations.

When we have an easy evaluation with $\mu = 0.9$ or $\mu = 0.75$, we get that $\mu_X \approx \sqrt{\hat{\mu}_Y}$ is a feasible estimation method. But with increasing difficulty, the pairwise consistency ratio between iterations decreases since an evaluation becomes more subject to arbitrary decisions when items are harder to evaluate. Thus the estimation method only shows good proximity when the restrictive assumptions for a two-choice human evaluation hold. Generally, the assumptions do not reflect a practical scenario for conducting a human evaluation for generative models since evaluation tasks will have varying difficulty levels, which need to be considered when analysing labelling consistency and performance. Thus, estimating the performance of two-choice human evaluation with sufficient data remains a part of future work.

4.1.8 Summary

The simulation experiments indicate that a highly probabilistic decision with our defined approach for all tested labelling strategies is achievable. Comparison between presented labelling strategies shows that assigning a single worker per request is a feasible evaluation method. In the following sec-
4.1. Agent-Based Human Evaluation

tions, we will further observe the proposed methodology by performing actual human evaluation experiments.
Figure 4.8: Maximum probability over evaluated requests computed with respect to the mean performance of One Worker.
4.1. Agent-Based Human Evaluation

Figure 4.9: Inter-project labelling consistency rates and mean pairwise proportion for given labelling methods when selecting $A$ as the better model.
4.2 Evaluation of Controlled Text Generation

The simulation gives insights into how our proposed decision strategy performs in the defined human evaluation setting. Simulating human evaluation comes with no cost, where we can abstract the evaluation process for multiple workers over an extensive set of requests. But eventually, one performs human evaluation with real workers evaluating generated data. A similar goal remains in a real evaluation setting. We want to make a confident decision when comparing two generative models. We want to analyse the performance of the proposed decision method and the required effort per labelling strategy in a crowdsourcing setting.

We focus on a single NLG domain for conducting human evaluation experiments by evaluating controlled text generation systems. We train different models such that there exists a noticeable difference for varying difficulty levels for model comparisons. In the following sections, we briefly introduce controlled text generation, followed by an overview of the experimental setup for the human evaluation and corresponding results.

4.2.1 Controlled Text Generation

A common goal for text generation applications is to augment datasets for supervised learning tasks in natural language processing. The main requirement for these applications is to support controllable text generation that enables systematic control for semantic and syntactic aspects of the generated text. Russo et al. recently proposed an NLG model called Control-Generate-Augment (CGA) that learns to control multiple semantic and syntactic attributes of a sentence with significant performance improvements in downstream NLP tasks [39]. We want to perform model comparisons of different difficulty levels for our human evaluation experiments to analyse the changes in required labelling effort between evaluation strategies. For that purpose, we use the CGA framework as a base for our human evaluation experiments since the implementation is publicly available\(^1\), which enables adjustments to create several variations of attribute-control text generation systems.

4.2.2 Experiment Setup

In the following sections, we will elaborate on the model selection for the human evaluation experiments trained with the CGA framework and the important design aspects of the crowdsourcing task.

\(^{1}\text{https://github.com/DS3Lab/control-generate-augment}\)
### 4.2. Evaluation of Controlled Text Generation

<table>
<thead>
<tr>
<th>Model</th>
<th>WD</th>
<th>Dataset Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_{ADV}$ + standard WD (V1)</td>
<td>0.3</td>
<td>~1300 sent.</td>
</tr>
<tr>
<td>$L_{ADV}$ + standard WD (V2)</td>
<td>0.7</td>
<td>~600.000 sent.</td>
</tr>
<tr>
<td>$L_{CGA}$ + cyclical WD (CGA)</td>
<td>$\zeta$</td>
<td>~600.000 sent.</td>
</tr>
</tbody>
</table>

Table 4.5: The configurations of key components in the CGA framework for three models (WD = word-dropout rate).

### Model Selection

Table 4.5 provides an overview of the architectural differences between three models trained on a dataset consisting of YELP business reviews\(^2\). The differences between the models are based on modifying key components required to implement the optimal version of CGA, such as with different losses and word-dropout routines.

The model $L_{CGA} + \text{cyclical WD}$ is trained using the configuration for the best version of CGA (which we refer to as CGA). The other two models, $L_{ADV} + \text{standard WD (V1)}$ and $L_{ADV} + \text{standard WD (V2)}$, are configured such they are expected to perform worse in comparison to CGA. The two models differ vastly in performance mainly due to different amount of data used during training, since V1 is only trained with $\sim 0.2\%$ of the available data. The loss, $L_{ADV}$, combines variational autoencoder (VAE) and a discriminator loss functions while the $L_{CGA}$ loss is $L_{ADV}$ combined with a context-aware loss function. The standard WD is based on the dropout routine applied in [8], which uses a fixed word dropout rate. In contrast, [39] apply cyclical WD, which represents a cyclical-dropout routine that computes dropout rate $\zeta$ according to current training iteration and initialised constraints. Further details related to the architecture components for CGA, such as the formal definition of the loss functions and the computations for the cyclical dropout routine, can be found in [39].

We construct two settings of comparisons based on empirical observations and automatic evaluation, where we provide examples of generated outputs from each model as well as the attribute matching accuracy in Appendix B. The first setting, V1 vs CGA, is relatively easy and the second one, V2 vs CGA, is more challenging but models are expected to be distinguishable. We refer two these two settings as major improvements and minor improvements, respectively.

### Evaluation Criteria

The models generate text based on semantic and syntactic attributes, and thus it is important to evaluate whether the sentences are generated ac-

\(^2\)https://github.com/shentianxiao/language-style-transfer
4. Results

cording to the provided attributes. Pre-trained classifiers can be used for automatic evaluation for attribute matching, while it remains difficult to automatically capture quality aspects of generated text [24]. Therefore, we focus on evaluating quality aspects for our experiment with human evaluation since the models must generate grammatically correct and coherent sentences. Also, considering quality aspects of the text enables reduction in provided description details when designing the crowdsourcing task since metrics like naturalness are not task-specific [34]. Recently, [48] provided recommendations for conducting human evaluation and emphasised the importance of focusing on single evaluation criteria per evaluation. Therefore we focus on one specific evaluation criteria in our experiments. We evaluate the **naturalness** of the generated sentences, which represents whether a given text could have been produced by a native speaker [34].

Data

As mentioned above, we trained three variations of attribute control models for our human evaluation experiments. We generated a total of 6000 sentences for each model for different combinations of the three following attributes: Verb tense, sentiment and person number. When preparing the pairwise comparison, sentences were paired on matching attributes and similar sentence length to reduce bias towards shorter or longer sentences. Repetitive sentences were removed from the samples of generated sentences to avoid having workers evaluating redundant sentences. The order of the sentences in each pair was randomised as well as the collection of configured sentence pairs. From each collection of sentence pairs, 500 random sentences were sampled to be published in a crowdsourcing setting for each experiment.

Crowdsourcing Setting

We used Amazon Mechanical Turk (AMT) to conduct the human evaluation experiments, a standard platform to collect evaluation from non-expert evaluators for various NLG tasks [12]. Jobs are posted on AMT as Human Intelligence Tasks (HITs), and thus the evaluation of a single request pair represents one HIT. The workers’ performance on the platform is measured with the number of approved HITs across different annotation experiments, where the requesters approve the HITs depending on the workers’ performance. The platform enables setting minimum qualifications of the workers to maintain the quality of collected annotations. Therefore to maintain quality control, we further increase the recommendation provided by [5] on qualification requirement on AMT for more reliable worker performance. All workers must have at least 10000 approved HITs and an approval rate greater than or equal to 98%. The location of the workers is required to be
in either the United States or Great Britain to ensure familiarity with the English language. In each experiment, we collected a total of 5000 evaluations for 500 sentence pairs, each evaluated by 10 random workers. The payment per comparison is $0.02 and the expected wage per worker was $9/hr. All workers that met the qualification requirements and participated in the evaluation were paid.

**Task Design**

The design of the evaluation is not task-specific, such that it could be adopted to another NLG domain. The aim was to design an interface to have minimum and clear instructions to avoid systematic confusion. Figure 4.10 shows an example of an evaluation task on Amazon Mechanical Turk.

![Figure 4.10: The task interface on Amazon Mechanical Turk.](image)

**4.2.3 Human Evaluation**

We aim to analyse the required labelling effort between different request difficulties in two experiments, (1) V1 vs CGA and (2) V2 vs CGA. For the more challenging comparison (V2 vs CGA), the experiment was executed on two distinct days to analyse the reliability of the labelling effort results.

To analyse collected human evaluations, we conduct a similar procedure as introduced in our simulation method to better represent the underlying distribution. We run 100 iterations over identical request pairs evaluated on AMT. We sample random workers without replacement for a single request in each iteration, depending on the given evaluation method. Note that due to the randomness present in the worker selection on AMT, where there is
4. Results

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Fleiss’ $\kappa$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCA vs V1</td>
<td>0.69</td>
</tr>
<tr>
<td>GCA vs V2 (R1)</td>
<td>0.27</td>
</tr>
<tr>
<td>GCA vs V2 (R2)</td>
<td>0.38</td>
</tr>
</tbody>
</table>

Table 4.6: Fleiss’ $\kappa$ to measure evaluators’ agreement.

no guarantee that the same worker evaluates multiple tasks, we omit the Fixed Worker method from our analysis with real human data.

4.2.4 Results

Overall, we reached the same decision of the preferred model with a high probability (0.9999) for all analysed evaluation methods. The least labelling effort is required with the One Worker evaluation method when evaluating models with major and minor improvement differences.

Major Improvements

An overview of the labelling effort required for a decision with increasing probability $1 - \delta$ when comparing CGA and V1 is presented in Table 4.7. We measure the agreement between collected annotations of the same items with Fleiss’ $\kappa$ [20], where the perfect agreement score equals 1, but the higher the score, the better the agreement among workers. The comparison between CGA and V1 yields a high consensus amongst evaluators shown with a high agreement score in Table 4.6 and minimal variation in the required labelling effort, especially for the methods which require several evaluators per request. That further indicates that there exists a common understanding for the goal of designed evaluation task. The majority voting methods 5 Workers and 7 Workers result in consistent labelling effort over the increasing probabilities of 0.99, 0.999, and 0.9999. But despite the consistent labelling effort, assigning each request to a single worker yields the least required labelling effort to conclude the better model over 100 iterations.

V1 was configured such that it results in a straightforward comparison with the optimal version of CGA by training with a minimal subset of the available training data. The consensus among the annotators and consistent labelling effort indicate that the tasks were not complex. Also as mentioned above, the design and purpose of the task, to select the more natural machine-generated sentence, was clear with little or no systemic confusion.

Minor Improvements

With a more difficult comparison, where the request pairs contain potentially more ambiguous selections, it is expected that the average required
4.2. Evaluation of Controlled Text Generation

<table>
<thead>
<tr>
<th>Methods</th>
<th>Avg. 99% CI</th>
<th>1 − δ = 0.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Workers</td>
<td>70 ± 0</td>
<td>70–70</td>
</tr>
<tr>
<td>5 Workers</td>
<td>50 ± 0</td>
<td>50–50</td>
</tr>
<tr>
<td>Max 3 Workers</td>
<td>30 ± 0</td>
<td>30–30</td>
</tr>
<tr>
<td>One Worker</td>
<td>11 ± 3</td>
<td>10–12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Avg. 99% CI</th>
<th>1 − δ = 0.999</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Workers</td>
<td>98 ± 0</td>
<td>98–98</td>
</tr>
<tr>
<td>5 Workers</td>
<td>70 ± 0</td>
<td>70–70</td>
</tr>
<tr>
<td>Max 3 Workers</td>
<td>43 ± 3</td>
<td>42–44</td>
</tr>
<tr>
<td>One Worker</td>
<td>19 ± 6</td>
<td>17–20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Avg. 99% CI</th>
<th>1 − δ = 0.9999</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Workers</td>
<td>133 ± 0</td>
<td>133–133</td>
</tr>
<tr>
<td>5 Workers</td>
<td>95 ± 0</td>
<td>95–95</td>
</tr>
<tr>
<td>Max 3 Workers</td>
<td>59 ± 4</td>
<td>58–60</td>
</tr>
<tr>
<td>One Worker</td>
<td>26 ± 8</td>
<td>25–28</td>
</tr>
</tbody>
</table>

Table 4.7: Labelling effort for model comparison: CGA vs V1. Confidence intervals are computed with bootstrap resampling with 99% confidence.

Labelling effort increases as examined in Section 4.1 compared to easier comparison tasks. Also, lower annotator agreement scores in Table 4.6 indicate less consensus for both repetitions than in the previous setting.

The required labelling effort over increasing probability for two separate human evaluations of V2 vs CGA, Repetition 1 (R1) and Repetition 2 (R2) are summarised in Table 4.8. Similar to previous findings, One Worker requires less labelling effort in comparison to all methods in both R1 and R2. For decisions with 0.99 probability, there is not a statistical significance between the required labelling effort between Max 3 Workers and One Worker in both R1 and R2 for the computed confidence intervals, but with increasing probability One Worker requires significantly less labelling effort in comparison to all methods in both R1 and R2.

Table 4.9 shows the ratio on how often One Worker achieve a decision over all iterations. The ratio is 1 for all other methods with given data in both repetitions. In R1, this ratio decreases with increasing probability since the method requires more request pairs beyond the provided data when supporting more confident decisions. That explains why the variance in R1 decreases opposite to the other methods since it lacks extended labelling efforts in several iterations. In contrast, R2 shows a consistent variance over increasing probability, although the same limitation occurs for 1 − δ = 0.9999.
Table 4.8: Labelling effort for two repetitions (R1, R2) of model comparison: CGA vs V2. Confidence intervals are computed with bootstrap resampling with 99% confidence. * represents that a decision was not achieved over all iterations with the collected sample of request pairs.

Table 4.9: The ratio (%) for R1 and R2 on achieved decisions for One Worker over 100 iterations when comparing CGA vs V2.
4.3 Evaluating End-to-End Text Generation

Our findings show that requiring a single label per request over a sufficient number of requests yields the same decision with the same probability as requiring multiple labels per request but with less labelling effort and thus less cost.

4.3 Evaluating End-to-End Text Generation

The human evaluation setup in the previous experiment is not task specific, where the focus is on evaluating quality aspects of the generated sentences. Therefore, we will re-use our evaluation setting to analyse model comparisons for a different natural language generation task. End-to-End (E2E) data-driven text generation is an important functionality for conversational agents where it has a significant impact on the user’s impression of dialogue systems [18].

Dataset called the E2E NLG dataset [33] was published for the first shared task for E2E NLG generation in spoken dialogue systems. A flat meaning representation (MR) represent the provided input data, containing attributes (that map to a specific datatypes) and corresponding values, and the systems generate sentences including all information present in the MR. An example of a data instance with MR and associated reference sentence is shown in Table 4.10. The participation in the shared task resulted in 62 submissions of systems applying various approaches, where the majority of the models followed a sequence-to-sequence (seq2seq) architecture. The results from the competition, as well as generated outputs from the top models, are publicly available. Therefore we perform a two-choice human evaluation with our proposed decision method to compare E2E NLG systems with available generated sentences. We examine if we can conclude the same optimal model with equal significance as reported with the competitions human evaluation approach.

4.3.1 Two-Choice Evaluation for Data-Driven Text Generation

Submitted systems were evaluated and ranked separately with automatic metrics and human evaluation. The human evaluations were collected with RankME (see Section 2.3.2), a relative magnitude estimation method to produce reliable rankings according to given quality criteria. Two separate human experiments were published in a crowdsourcing environment each focused on a different metric, naturalness and overall quality. The overall quality requires workers to evaluate grammatical correctness and the correctness of the generated sentence concerning the provided MR, thus specific for this

text generation task. Naturalness has the exact definition as presented in Section 4.2.2 where sentences were shown to the workers without the original input data.

Two-choice evaluation shares a core element with the RankME method. Both methods use relative comparisons to state which sentence is better for each request. The main differences are that RankME allows ties, collect scores with magnitude estimation and focuses on evaluating multiple systems (at least three) simultaneously. Next, we will discuss our experimental setup for evaluating E2E NLG systems with our approach, followed by a discussion regarding our collected results in contrast to the rankings produced in the original competition.

### 4.3.2 Experiment Setup

We select two models for our experiment: Slug2Slug and Sheffield2. The system Slug2Slug is a seq2seq ensemble model with LSTM/CNN encoders and LSTM decoder and applies heuristic slot aligning re-ranking and data augmentation, while Sheffield2 is a simple vanilla seq2seq. Slug2Slug yielded the best performance for most metrics in an automatic evaluation and ranks as the optimal model according to the primary human evaluation metric, the overall quality. For naturalness, Slug2Slug was the second-best model. Sheffield2 was among the worst-performing models regarding overall quality but resulted in being the top model according to naturalness. Thus, we expect noticeable differences between these models and will perform evaluation with available generated data.

For each model, we have in total of 630 MR and corresponding generated sentences. Examples of generated sentences for both models are shown in Tables 4.11 and 4.12. Similar to our previous experiment, we generate request pairs based on the provided input data, remove identical sentences from the collection (as well as pairs that resulted in identical sentences), and randomise the order of each request pair. From the configured sentence pairs, we sampled 500 random request pairs to publish on AMT.
4.3. Evaluating End-to-End Text Generation

**Input Data and Generated Sentence**

**I.** name[Giraffe], eatType[pub], food[Fast food], familyFriendly[yes], near[Rainbow Vegetarian Café]

**O.** Giraffe is a family friendly pub in the riverside area near Rainbow Vegetarian Café.

**I.** name[The Punter], eatType[restaurant], food[English], priceRange[£20-25], area[riverside], familyFriendly[yes], near[Raja Indian Cuisine]

**O.** The Punter is a kid friendly English restaurant in the riverside area near Raja Indian Cuisine with a price range of £20-25.

**I.** name[The Phoenix], eatType[pub], food[French], priceRange[£20-25], area[riverside], familyFriendly[yes], near[Raja Indian Cuisine]

**O.** The Phoenix is a kid friendly French pub in the riverside area near Raja Indian Cuisine with a price range of £20-25.

Table 4.11: Examples of generated sentences with model *Slug2Slug* according to corresponding input data.

---

**Input Data and Generated Sentence**

**I.** name[Giraffe], eatType[pub], food[Fast food], area[riverside], familyFriendly[yes], near[Rainbow Vegetarian Café]

**O.** Giraffe is a family friendly pub located near the Rainbow Vegetarian Café.

**I.** name[The Punter], eatType[restaurant], food[English], priceRange[£20-25], area[riverside], familyFriendly[yes], near[Raja Indian Cuisine]

**O.** The Punter is a family friendly restaurant near the Raja Indian Cuisine. It serves English food and is in the riverside area.

**I.** name[The Phoenix], eatType[pub], food[French], priceRange[£20-25], area[riverside], familyFriendly[yes], near[Raja Indian Cuisine]

**O.** The Phoenix is a family friendly pub located near the Raja Indian Cuisine.

Table 4.12: Examples of generated sentences with model *Sheffield2* according to corresponding input data.
4. Results

The evaluation design is not task-specific. Thus, we evaluate the models according to the naturalness criteria without making any modifications to the original task design for AMT. We publish the evaluation on AMT with the same worker qualification and the same number of collected human evaluations (5000 in total) as presented in Section 4.2.2.

We publish a single comparison experiment, where we evaluate Slug2Slug vs Sheffield2 based on naturalness. To analyse the collected evaluations, we run 100 iterations on identical sentence pairs with varying workers to represent the underlying distribution. We investigate whether we can make the same decision as reported in [19] according to naturalness with two-choice evaluation associated with Hoeffding inequality.

4.3.3 Results

Performing two-choice human evaluation with a corresponding decision approach resulted in the systems being indistinguishable for all labelling strategies as shown in Figure 4.11. In contrast, Sheffield2 is reported significantly better (with a p-level of $p \leq 0.05$) than Slug2Slug when comparing the models for naturalness in the original shared task.

There can be several reasons for these differences and why the models are indistinguishable in our setting. First, five random systems (21 systems in total) were evaluated simultaneously for the competition evaluation, and 4239 data points were collected. From these data points, 42390 pairwise comparisons were produced or roughly 2018 comparisons per system. These pairwise comparisons are used to make the final rankings using the TrueSkill algorithm (see Section 2.3.2) with bootstrap resampling. Thus comparisons to a broader range of systems seem to support creating more significant rankings. Secondly, the sentences are very structured with respect to provided input, and therefore the differences are mainly noticeable in minor structure differences of the sentences. We also noticed that sentences generated by Sheffield2 are generally shorter than sentences generated with Slug2Slug since Sheffield2 sometimes lacks required information. Workers can tend to prefer longer or shorter sentences overall which can also impact the results. Analysing the worker performances, workers sometimes showed a clear preference towards selecting longer or shorter sentences overall. But disagreements among workers on the better model can be confirmed by a very low agreement rate of Fleiss $\kappa = -0.01$. The observed agreement among the annotators was lower than the expected agreement by chance, further supporting that the models are difficult to separate. Third, according to our decision approach with Hoeffding inequality, the provided test data contains too few sentences to conclude the better model with significant probability, as shown in Figure 4.12 when evaluating the systems with a single random worker per request pair.
4.3. Evaluating End-to-End Text Generation

Figure 4.11: Mean proportion with associated decision bounds with probability $1 - \delta = 0.99$ for evaluating Slug2Slug vs Sheffield2.

Figure 4.12: Maximum probability over increasing number of requests for Slug2Slug vs Sheffield with the One Worker labelling strategy.
Simulating human evaluation enables observation of different labelling strategies without cost, which gives valuable insight before actual evaluation with crowdsourcing workers. But the simulation only relies on probabilistic distributions and does not reflect every human behavioural aspect, and thus it is also essential to explore actual human evaluation.

Comparing the results in the simulated and real human evaluation experiments show similar trends. The simulation results in Section 4.1 and the human evaluation in Section 4.2 yield that assigning a single worker per request requires the least labelling effort in comparison to labelling strategies that require multiple workers per request. In Section 4.3.1, we explored a different NLG domain where the provided sample size was not sufficient for a confident conclusion. Analysing the observed data and the proportion of the selected model indicates that the comparison between models is too hard, which is a scenario that can occur when comparing models with similar performance according to given criteria.

Additionally, we can observe from the simulation experiment where we have the most varying worker capability for the most challenging evaluation experiment ($\mu = 0.0625$) that the standard deviation is larger for a Fixed Worker in comparison to 5 Workers. That can indicate a significant outlier when recording the required labelling effort over multiple iterations. A worker with capability $c = 0.5$, makes evaluations across request pairs with $\mu = 0.0625$ close to random on average. Thus, such evaluation will need a large set of request pairs to converge. Although we can reach a decision with Hoeffding with the required number of requests, the effort for too similar models will probably exceed what is considered practical. Thus one might want to consider defining an upper bound for the labelling effort or the number of requests to specify indistinguishable models, which could be explored as future work. Furthermore, one might want to estimate what causes models to be indistinguishable.
Another limitation observed from analysing the labelling effort is the large standard deviation across experiments. Similar to the discussion above, we believe that large standard deviation results from measured outliers. The outliers can occur because of low or high worker capabilities sampled over given labelling strategies during the evaluation. Hoeffding only takes the average performance into account when bounding the probability. Still, for future work, other techniques could be explored to bound the variance of the performance and how it associates with the required labelling effort. That could further support a better understanding of the labelling effort distributions.

Assigning a single worker per request pair goes against the recommendations provided by van der Lee et al. [48]. But Khashabi et al. found that collecting one label per instance resulted in less variance compared to majority voting strategies when computing leaderboard scores with human evaluation [28], which aligns with our results. That further supports that assigning a single random worker per request deserves more attention when conducting a human evaluation.
Chapter 6

Conclusion

In this thesis, we proposed a methodology to decide the better model with high probability when comparing two generative models with two-choice human evaluation. To examine the presented approach, we designed a simulation framework to analyse the performance of simulated human judgements when selecting the better outputs through series of evaluation tasks. We experimented with multiple labelling strategies that differ in worker capabilities and the number of judges assigned per request to assess the needed labelling effort. With insights from our simulation, we conducted a real human evaluation with crowdworkers on Amazon Mechanical Turk to evaluate natural language generation models. We showed that we could make confident decisions across all labelling strategies with collected human judgements by applying the proposed decision method. Both simulated and real human evaluation experiments showed that recruiting a single worker per request required the least labelling effort to reach a high probabilistic decision on the better model with relative model comparisons. Furthermore, assigning a different worker per request enables trivial parallelization such that less time is required for evaluation.

The proposed method is the first option of its kind to analyse the required labelling effort for different labelling strategies with a defined target probability. The method can enable the design of better evaluation strategies to require less human effort when selecting the better NLG model with high confidence.
Appendix A

Extended Simulation Analysis

The following section contains additional simulation results when simulating two-choice human evaluation.

A.1 Decreasing Probability

Figure A.1 shows the change in labelling effort over decreasing probability $1 - \delta$ for $\mu = 0.25$. The shaded area shows the associated standard deviation of the labelling effort.
Figure A.1: Average labelling effort with standard deviation over decreasing probability $1 - \delta$ with $\mu = 0.25$. 

58
The following sections contain extended information for selecting attribute control models before configuring human evaluation experiments in a crowdsourcing setting.

### B.1 Automatic Evaluation

Table B.1 shows attribute matching accuracy for all generated sentences by each evaluated model. Note that the rightmost column indicates an unexpected low accuracy for the person number attribute compared to the results reported in [39]. Still, low person number accuracy has little or no impact on the configuration of defined comparison settings.

<table>
<thead>
<tr>
<th>Model</th>
<th>Sentiment</th>
<th>Tense</th>
<th>Person</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>65.60%</td>
<td>39.48%</td>
<td>41.03%</td>
</tr>
<tr>
<td>V2</td>
<td>95.93%</td>
<td>96.53%</td>
<td>56.53%</td>
</tr>
<tr>
<td>CGA</td>
<td>98.68%</td>
<td>98.08%</td>
<td>56.02%</td>
</tr>
</tbody>
</table>

Table B.1: Attribute matching accuracy (in %) of 6K generated sentences for each evaluated model.
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Present / Positive / Plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>There are closed.</td>
</tr>
<tr>
<td>V2</td>
<td>The rooms are clean and nicely appointed.</td>
</tr>
<tr>
<td>GCA</td>
<td>They have a great selection of beers and they are always friendly.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Present / Positive / Singular</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>I am always packed.</td>
</tr>
<tr>
<td>V2</td>
<td>Everything else is great.</td>
</tr>
<tr>
<td>GCA</td>
<td>The food here is always good.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Past / Positive / Plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>The first time was packed.</td>
</tr>
<tr>
<td>V2</td>
<td>All of the steaks were great.</td>
</tr>
<tr>
<td>GCA</td>
<td>This was my favorite restaurants.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Past / Positive / Singular</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>Oh and the food.</td>
</tr>
<tr>
<td>V2</td>
<td>He also was very good.</td>
</tr>
<tr>
<td>GCA</td>
<td>The best i had in phoenix.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Present / Negative / Plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>Nothing is awesome.</td>
</tr>
<tr>
<td>V2</td>
<td>They are better than you.</td>
</tr>
<tr>
<td>GCA</td>
<td>Worst wings i have ever had.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Present / Negative / Singular</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>But i am going to.</td>
</tr>
<tr>
<td>V2</td>
<td>Do not waste your time here.</td>
</tr>
<tr>
<td>GCA</td>
<td>This is a very expensive hotel.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Past / Negative / Plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>There were incredibly cold.</td>
</tr>
<tr>
<td>V2</td>
<td>The people that used to be the other reviews.</td>
</tr>
<tr>
<td>GCA</td>
<td>We were not happy with the food.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Past / Negative / Singular</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>Money at this place.</td>
</tr>
<tr>
<td>V2</td>
<td>I just went to the drive-thru and the service.</td>
</tr>
<tr>
<td>GCA</td>
<td>The waiter did not know what i wanted to pay for a drink.</td>
</tr>
</tbody>
</table>

Table B.2: Examples of generated sentences from models V1, V2, and CGA, according to variations of three input attributes (tense, sentiment and pronoun).
Bibliography


[7] Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn,


[28] Daniel Khashabi, Gabriel Stanovsky, Jonathan Bragg, Nicholas Lourie, Jungo Kasai, Yejin Choi, Noah A. Smith, and Daniel S. Weld. GE-
Bibliography


Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor’s thesis, Master’s thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):
Understanding Human Potentials for Evaluating Generative Models

Authored by (in block letters):
Thorleiksdóttir Thórhildur

Name(s): First name(s):

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the ‘Citation etiquette’ information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date: Signature(s):
Zürich, 18.08.2021 Perúlfur Pórleiksóttir

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.