Comprehensive Evaluation of Machine Learning Experiments

Algorithm Comparison, Algorithm Performance and Inferential Reproducibility

Inauguraldissertation zur Erlangung des akademischen Grades
Doktor der Philosophie der Neuphilologischen Fakultät
der Ruprecht-Karls-Universität Heidelberg

vorgelegt von

Michael Hagmann

am
07. August 2023

Hierbei handelt es sich um eine Heidelberger Dissertation.

Erstgutachter:	Prof. Dr. Stefan Riezler
	Institut für Computerlinguistik, Universität Heidelberg
Zweitgutachter:	Prof. Dr. Katja Markert
	Institut für Computerlinguistik, Universität Heidelberg

Abstract

This doctoral thesis addresses critical methodological aspects within machine learning experimentation, focusing on enhancing the evaluation and analysis of algorithm performance. The established "train-dev-test paradigm" commonly guides machine learning practitioners, involving nested optimization processes to optimize model parameters and meta-parameters and benchmarking against test data. However, this paradigm overlooks crucial aspects, such as algorithm variability and the intricate relationship between algorithm performance and meta-parameters. This work introduces a comprehensive framework that employs statistical techniques to bridge these gaps, advancing the methodological standards in empirical machine learning research. The foundational premise of this thesis lies in differentiating between algorithms and classifiers, recognizing that an algorithm may yield multiple classifiers due to inherent stochasticity or design choices. Consequently, algorithm performance becomes inherently probabilistic and cannot be captured by a single metric. The contributions of this work are structured around three core themes:

Algorithm Comparison: A fundamental aim of empirical machine learning research is algorithm comparison. To this end, the thesis proposes utilizing Linear Mixed Effects Models (LMEMs) for analyzing evaluation data. LMEMs offer distinct advantages by accommodating complex data structures beyond the typical independent and identically distributed (iid) assumption. Thus LMEMs enable a holistic analysis of algorithm instances and facilitate the construction of nuanced conditional models of expected risk, supporting algorithm comparisons based on diverse data properties.

Algorithm Performance Analysis: Contemporary evaluation practices often treat algorithms and classifiers as black boxes, hindering insights into their performance and parameter dependencies. Leveraging LMEMs, specifically implementing Variance Component Analysis, the thesis introduces methods from psychometrics to quantify algorithm performance homogeneity (reliability) and assess the influence of meta-parameters on

performance. The flexibility of LMEMs allows a granular analysis of this relationship and extends these techniques to analyze data annotation processes linked to algorithm performance.

Inferential Reproducibility: Building upon the preceding chapters, this section showcases a unified approach to analyze machine learning experiments comprehensively. By leveraging the full range of generated model instances, the analysis provides a nuanced understanding of competing algorithms. The outcomes offer implementation guidelines for algorithmic modifications and consolidate incongruent findings across diverse datasets, contributing to a coherent empirical perspective on algorithmic effects.

This work underscores the significance of addressing algorithmic variability, meta-parameter impact, and the probabilistic nature of algorithm performance. This thesis aims to enhance machine learning experiments' transparency, reproducibility, and interpretability by introducing robust statistical methodologies facilitating extensive empirical analysis. It extends beyond conventional guidelines, offering a principled approach to advance the understanding and evaluation of algorithms in the evolving landscape of machine learning and data science.

Contents

1 Introduction 1

1.1 Research Question and Contribution . . . 2

2 Algorithm Comparison 7

2.1 The Principles of Statistical Hypothesis Testing . . . 13

2.1.1 (Re-)Sampling-Based Significance Tests I: Bootstrapping . . . 18

2.1.2 (Re-)Sampling-Based Significance Tests II: Permutation . . . 20

2.1.3 Excursion: Multiple Comparisons and Elementary Post-hoc Procedures . . . 23

2.2 Model-Based Algorithm Comparison: Toolbox . . . 25

2.2.1 Linear Mixed Effects Models: Specification Example . . . 28

2.2.2 Linear Mixed Effects Models: Fitting a Model to Data (Parameter Optimization) . . . 32

2.2.3 Linear Mixed Effects Models: Statistical Inference (Likelihood Ratio Statistic) . . . 33

2.3 Model Based Algorithm Comparison: Analyzing an Example . . . 40

3 Algorithm Performance Analysis 55

3.1 Untangling Terminology: Reliability, Agreement etc. . . . 57

3.2 A Unifying View on Algorithm Performance and Annotation . . . 58

3.3 State of the Art Methods for Annotation and Algorithm Performance Analysis . . . 59

3.3.1 Agreement Coefficients for Data Annotation . . . 60

3.3.2 Bootstrap Confidence Intervals for Model Evaluation . . . 65

3.4 Bridging the Gap: Model-based Reliability Analysis . . . 69

3.4.1 The Basic Principles of Variance Component Analysis (VCA) . . . 70

3.4.2 VCA based Reliability Coefficients . . . 73

3.4.3 Data Annotation Analysis . . . 79

3.4.4 Algorithm Performance Analysis . . . 82

4 Inferential Reproducibility 95
4.1 A Scheme for Analyzing Inferential Reproducibility 97
4.2 Linear Mixed Effects Models 98
4.3 Generalized Likelihood Ratio Tests w/ and w/o Measurement Variation . 99
4.4 Variance Component Analysis and Reliability Coefficients 101
4.5 A Worked-Through Example 103
4.6 Related Work 108
4.7 Discussion 108

5 Concluding Remarks 111

List of Figures 115

List of Tables 117

Bibliography 119

Chapter 1

Introduction

Machine learning is a research field that has been explored for several decades and has recently started to affect many areas of modern life. Machine learning aims to construct learning algorithms that, based on input-output examples or alternative feedback mechanisms, output a mathematical function (represented by a computer program). The learning process of this algorithm is essentially an optimization problem aiming to minimize a loss objective, and its solution relays on methods of mathematical optimization [BCN18]. The loss objective is typically based on a sample of relevant input-output examples (training data) and is called *empirical risk objective*. The minimization of this empirical risk objective is only a means to an end, namely to construct an algorithm that can create functions whose expected loss over the $(\text{input}, \text{output})$-distribution is minimal. The Machine learning practitioner has to resort to this indirect practice because the expected risk is not accessible as it requires exact knowledge of the principally unknown probability law of $(\text{input}, \text{output})$. The relation between empirical risk minimization and expected risk is studied in statistical learning theory [BBL04; LS11; Vap98]. Statistical learning theory thus provides the inductive principles sustaining the learning process.

The findings of statistical learning theory directly impact the workflow of machine learning experiments in natural language processing (NLP) and data science. Empirical research in these areas generally starts with a collection of exemplary input-output pairs –partitioned into three disjoint subsets called training, development, and testing data– assumed to represent independently drawn samples from the same data generating probability space (so-called i.i.d. samples). This property is enforced and conserved by techniques like random shuffling between splits [Arj+19] or experience replay [Sch19] and is a condicio sine qua non for most consistency guarantees [Vap98; LS11]. The typical workflow of a machine learning experiment motivated by statistical learning theory can be phrased as a nested optimization process where the model parameters are optimized on given training data, and the meta-parameters are tuned on development data. Finally, the optimized model is applied to a benchmark test data set to obtain an unbiased estimation of its expected risk. We will call this scheme the *train-dev-test paradigm* of NLP and data science.[1]

The train-dev-test paradigm assigns the machine learning practitioner the task of improving model performance, limited only by the computational resources at her hand to (re-)train complex models under extensive exploration of meta-parameter configurations and her experience in skillfully applying a range of technical stratagems. Thus she does not need to concern herself with questions about the data compilation process, what the machine learning model has learned, or how the learning process is systematically influenced by diverse sources of variability attached to the implementation of model training. These questions are not investigated by classical statistical learning theory and must be addressed in the empirical analysis of machine learning experiments.

In this work, I advocate that the investigation of algorithm performance variation should be an integral part of machine learning methodology and addressed by the analysis of an experiment. The current discussion of methodological issues and standards in empirical machine learning research is at the state of informal guidance by Dos and Don'ts [BD21; Lon21] at best compiled into checklists. My principle goal is to address methodological shortcomings of the train-dev-test paradigm[2] by providing adequate statistical methods allowing the machine learning practitioner to analyze them in her work.

1.1 Research Question and Contribution

This thesis is a cumulative effort. Thus all parts of it have been previously published in peer-reviewed publications, co-authored with my supervisor. The excurse on multiple testing in chapter 2 has previously appeared in my master's thesis [Hag16].

The fundamental notion sustaining all contributions of this thesis is the distinction between algorithms and classifiers which are specific instances of an algorithm [Die98]. This distinction is often blurred in the machine learning literature by an ambiguous usage of the term "model" or other synonyms that can mean both. This lack of conceptual

[1] Clearly, this paradigm is pervasive in machine learning and artificial intelligence in general, for example, in the area of image processing that uses similar methods and exhibits similar problems as the area of natural language processing.

[2] No matter if it is founded in classical statistical learning theory or more recent approaches [KKB20; Arj+19; She+21].

clarity has led to an unclear situation regarding evaluating machine learning experiments. Contemporary practice implies that comparing particular algorithm instances can be a valid substitute for algorithm comparison. This identification is questionable as soon as the involved algorithms outputted classifier is not unique for a given training data. Unfortunately, this is the case for almost all contemporary algorithms, especially when they employ a deep neural architecture. There are multiple reasons, e.g., a non-convex optimization problem specified by the algorithm, resorting to stochastic optimization methods, the deliberate usage of randomness to improve algorithm performance, or unspecified parameters in the algorithm description. Hence, we need to think of an algorithm as a collection of potential classifiers that can be outputted depending on the exact choice of implementation details. Thus, by extension, algorithm performance can not be characterized by a single number but must instead be captured by a distribution over evaluation scores.

The contributions of this thesis can be grouped into three interrelated efforts aiming at an improved evaluation of machine learning experiments that meet the standard of other knowledge-oriented empirical sciences. It differs from current attempts to increase the reproducibility of machine learning experiments by promoting analytical methods vs. reporting standards and advocates accepting the in-determinism inherent in contemporary machine learning methods and not eliminating it. Nevertheless, the methods presented in this thesis aim to avoid replacing these attempts but to amend them and add to the discussion. To this end, I focus on the following topics.

Algorithm Comparison. Undoubtedly, comparing algorithms is the primary purpose of most empirically oriented machine learning research. In this chapter, I propose the usage of LMEM to analyze the evaluation data of a machine learning experiment. This approach offers several advantages to previous related work. Most notably, the ability of LMEMs to model complex data structures that deviate from the ubiquitous iid assumption allows simultaneous analysis of all the instances of the algorithms under investigation. Secondly, LMEMs allow building complex conditional models of the expected risk[3] of an algorithm, thus enabling a comparison of algorithms dependent on arbitrary data properties. Parts of the work presented in this chapter were previously published in the chapter "Significance" of [RH21]. I reused the following sections, which the author of this work mainly contributed:

- Exposition to the principles of hypothesis testing.

[3] The expected value out-of-sample value of an evaluation metric.

- Exposition of LMEM theory.
- Showcase analysis of Kreutzer et al. data set.

Algorithm Performance Analysis. One shortcoming of the contemporary state-of-the-art evaluation of machine learning experiments is its tendency to treat and analyze algorithms and classifiers in a black-box fashion. This attitude prohibits any empirically grounded insights into the relationship between an algorithm's performance concerning a task and the choice of its meta-parameters. Based on an LMEM implementation of Variance Component Analysis, I demonstrate how notions originated in psychometrics can be adapted to quantify the performance homogeneity of an algorithm (called reliability) and asses the influence of specific meta-parameters on algorithm performance. I will also show how the flexibility of LMEMs to build models of the expected risk conditional on arbitrary data properties allows a fine-grained analysis of this dependency. Another novel aspect of this chapter is the argument that algorithm performance and data annotation analysis can be investigated using the same analytical tools. This aspect is particularly interesting when the algorithm depends on human feedback, e.g., [KBR20] so that the mechanism to collect this feedback can be seen as a part of it.

Parts of the work presented in this chapter were previously published in the chapter "Reliability" of [RH21]. I reused the following sections, which the author of this work mainly contributed:

- Exposition of the current state-of-the-art.
- Exposition to the principles of G-Theory.
- Exposition to the principles of Variance Component Analysis.
- Showcase analysis of Kreutzer et al. data for Annotation and Algorithm Performance Analysis.

Inferential Reproducibility. This chapter shows how the techniques and methods presented in the previous chapters can be combined to analyze a machine learning experiment. This analysis uses every model instance generated during the experiment to gain a more nuanced picture of the competing algorithms. The benefits of such an analysis are not only implementation guidelines that stem from a detailed understanding of the conditions when one algorithm is superior to the other or which meta-parameter is crucial but also a more coherent picture that allows integrating incongruent findings about an algorithm on different data sets into a cohesive picture throwing empirically grounded light on the effects of the proposed algorithmic modification. The material of this chapter was previously published as [HR21].

Furthermore, I provide R and Python code to replicate the demonstrated examples and facilitate easily adapting the presented methods to the NLP and data science community. Code and data are available at `https://www.cl.uni-heidelberg.de/statnlpgroup/empirical_methods/`

Chapter 2

Algorithm Comparison

Comparing learning algorithms is the central matter of most machine learning experiments in academic machine learning research and is pivotal to demonstrating algorithmic advancements. Given its importance in justifying progress in empirically orientated machine learning research, this topic has received moderate attention from the machine learning community.

A first rudimentary taxonomy on the kind of comparisons that occur in machine learning experiments was proposed by [Die98]. This taxonomy was a first attempt to match typical analytical questions of machine learning experiments to appropriate statistic methods[1] (if available at the time) for classification tasks. A still important criterion of this taxonomy is the clear distinction between evaluating a *classifier* and a *learning algorithm*. A classifier[2] is a function that, given an input example, outputs (or predicts) a class label for this input example. In contrast, a learning algorithm is a function that constructs a classifier given a collection of training examples with known labels.

Unfortunately, this essential initial work has found no echo in contemporary benchmark experiment analysis where systematic uncertainty estimation is a neglected problem [FP19], and statistical inference procedures established in a broad spectrum of empirical sciences, e.g., hypothesis tests, are often entirely ignored. Instead, machine learning researchers waive proper methods in favor of rules of thumb for sufficiently large differences between observed evaluation scores. Thresholds are usually provided by tradition in respective application areas and are seldom justified by rigorous arguments. For example, in the area of machine translation, result differences of at least $1-2$ BLEU points [Pap+02] seem to be publication-worthy and are often termed "significant" [MFR21]

[1] This term encompasses appropriate experimental designs along with formal techniques to analyze them.

[2] Henceforth, I use this term in a broader meaning and use it as a synonym for a trained model, model instance, etc.

invoking associations with statistical hypothesis testing. A closer look at the current standard of experiment evaluation reveals an even bleaker fact. Let us take the area of machine translation, for example. The de facto standard experimental scenario can be sketched as follows: First, obtain a collection of classifiers for each algorithm in the comparison by re-executing each with different meta-parameter configurations on the available training data and select exactly one representative classifier for each algorithm. Typically this selection is made by picking the classifier that performs best on development data for each learning algorithm. These representatives called the *best* models, are then applied to the test set and compared descriptively in an above-described manner. Thus the comparison of algorithms is implicitly and wrongly reduced to classifier comparison.

Indubitable academic research is interested in learning algorithms and not specific classifiers. The abstract research question that motivates most academic machine learning experiments is:

> Given two learning algorithms A_1 and A_2, which algorithm will produce more accurate classifiers[3] for a given task and training data[4] of size N?

The provision of a proper statistical method –an experimental design that specifies which data needs to be collected and a suitable device for statistical inference– to answer this question is not as straight as it seems prima facie. The critical issue that prevents the direct and unconditional application of standard statistical comparisons techniques like the t-test, analysis of variance (ANOVA), or their non-parametric or resampling-based variants to analyze this question is the fact that the outcome of a machine learning experiment is affected by a multitude of decisions needed to implement a learning algorithm in a computer program. The most obvious are algorithm parameters inherent, like widths and numbers of the hidden layers of a deep neural network (DNN), the penalization weight for penalized learning objectives but also includes more implementation-related details like choice of random seeds for stochastic optimization methods, the usage of generalization enhancing stochastic methods like drop-out, library versions, etc. In the following, I will summarize all these choices not explicitly addressed by an algorithm under the umbrella term *meta-parameter*. Conceptually we can think of meta-parameters as random variables to capture the uncertainty associated with their choice. Hence, the outputted classifier of an algorithm and, thereby, the evaluation metric is a random variable even when

[3] Here the term classifier is to be understood in a very general meaning and is a synonym for a concrete model instance or trained model.

[4] Typically data sets from different domains are used during an experiment.

the training data is regarded as non-random[5]. Thus a statistical inference procedure suitable to analyze the experimental data from machine learning experiments must be able to incorporate randomness due to training data, meta-parameter, and test data[6] Standard statistical comparison techniques usually deal only with the latter, namely the uncertainty of the out-of-sample risk estimator due to test set sampling. Thus to apply them, the machine learning researcher has to restrict her comparison to *one* classifier per algorithm. This representative is usually found by executing an algorithm several times with different meta-parameter values on the same training set and selecting[7] the classifier with the best out-of-sample risk estimator based on another set of input-output examples called *development set*. This proceeding has severe implications for the analysis. Firstly, it reduces the comparison of algorithms to the comparison of classifiers which is a significant constraint to the above-stated research question. Secondly, such an analysis treats the training data and the meta-parameters of the chosen model as given, thereby ignoring any uncertainty associated with these entities. Thus the conclusion drawn from such an analysis does not match the above question. The correct corresponding research question can is:

> Given two learning algorithms A_1 and A_2 with predefined meta-parameter configurations and given training data, which algorithm will produce the best classifier?

The subject of this question deviates severely from the original, and its scope is significantly limited compared to the original. Nevertheless, this research direction received much attention when research tried to match standard statistical tests to popular NLP evaluation metrics [Hot+05], [Dro+20]. To lift some of the shortcomings of this approach, researchers suggested using multiple testing procedures and meta-analytical methods to synthesize the hypothesis test statistics from several similar studies or data sets within one study into a joint overall test [Dro+17]. But the fundamental problem that this kind of algorithm comparison doesn't provide an answer to the actual research question was not and can not be solved.

Several authors try to capture meta-parameter uncertainty by applying the U-test [RG18], the Kolmogorov–Smirnov test([RG17]), or hypothesis test inspired procedures

[5] Actually this assumption would contradict the research interest as formulated above and limit the scope of the conclusions that can be drawn from the experiment.

[6] A set of input/output examples disjoint from the training which is used to estimate the out-of-sample risk of a classifier.

[7] Most commonly machine learning researches neglect estimator uncertainty due to development set sampling for this purpose.

[DSR19] to so-called *score distributions.* A score distribution for a learning algorithm is generated by executing it several times with different meta-parameter values on the same (fixed) training data and estimating a summary evaluation metric for each resulting classifier based on the (same fixed) test set. The collection of these (point) estimates is called the score distribution of an algorithm. While such an approach accounts for meta-parameter uncertainty, it completely neglects that each classifier's estimated summary evaluation metric is based on sampled test data and hence a random variable contribution to overall uncertainty. Consequently, score distribution-based techniques underestimate the variability of the evaluation scores and therefore have an elevated and non-controlled type-I error probability. A second problem of this approach is that score distribution-based tests don't leverage the full potential of the experimental design. Namely, they don't take advantage of the fact that the algorithms are evaluated on the same test data and thus constitute repeated measurements of the same object, a fact that, when taken into account, usually dramatically increases the power of a hypothesis test. Besides this, severe technical issues ignoring the test set and estimator variability limit the inference's scope and don't allow conclusions beyond the given test sample. In summary, inference based on this approach must be interpreted concerning the following research question:

> Given two learning algorithms A_1 and A_2 as well as training data which algorithm will produce better classifiers for a given test set?

Nevertheless, the merit of this branch of research is that it emphasizes the need to take several instances (outputted classifiers) of an algorithm into account for a proper algorithm (not classifier) comparison. Consequently, authors promoting it [Bou+21] stress that as many sources of variation as possible should be randomized during an experiment to gain a comprehensive comparison. In this spirit, relatively recent work [Bou+21] revived ideas [Hot+05] developed for rather elementary machine learning models whose learning process can be captured by a convex optimization problem without meta-parameters. Back then, the main question was to capture the evaluation metric variation due to data sampling when comparing algorithms. The most obvious idea is to randomly and evenly break a sufficiently large data sample randomly and evenly into k distinct training and test data pairs and train and evaluate an algorithm on each pair to collect the evaluation data, which is then analyzed similarly to score distribution data. If splitting the original sample is not feasible, several resampling schemes based on bootstrapping or cross-validation to construct appropriate variance estimators have been investigated. A technical intricacy of cross validation schemes is the fact that they produce overlapping (resampled) data sets. Consequently, this fact induces a complex correlation structure among the resulting evaluation statistics (scores). [BG04] showed that there exists no universal unbiased estimator of the variance for cross validation schemes and that the naive variance estimator grossly underestimate the true variance, thus yielding a severely inflated (larger than nominal) type-I error rate if one applies a t-test to this data. With bootstrapping the situation is even more complicated. As a matter of fact the term refers to a wide range of techniques whose common notion is to replace the true but unknown data distribution F_X by an empirical estimate $\hat{F}_{X_n}$ based on a sample X_n and then do probabilistic calculations based on (re-)drawing samples of the same size as X_n from $\hat{F}_{X_n}$. Thus all inference is conditional on the original sample X_n. Depending on the exact specifics of the score statistics[8], its asymptotic (large sample) behavior and the desired inference the bootstrap can work or not. Examples were the nearly ubiquitously used non-parametric bootstrap, that is resampling the original sample with replacement, fails nearly completely are discussed in [BF81; LH23; HHS93; Che11] and a technical discussion of this topic can be found here [Hal13]. In summary, there exist no universal theorem that guarantees the soundness of any bootstrap procedures under all possible conditions.

Another exciting question not raised explicitly by academic research yet asks if an observed performance gain of an algorithm or a classifier constructed by a newly developed algorithm is homogeneous over its domain. Or, in other words, can one characterizes inputs that benefit more than others from the newly developed algorithm? To this end, one needs to build conditional models of the expected risk (performance), something classical statistical hypothesis tests and construction principles are not designed for.

One of my research goals is to promote linear mixed effect model (LMEM) based experiment analysis which circumvents the above sketch shortcomings of alternative approaches. LMEMs represent one of the most flexible classes of regression models. In its broadest abstraction, a regression function links a particular deterministic feature C of a random variable Y to some determining factors X. Mathematically this is expressed by

$$C\left(Y|X\right) = g(X).$$

Restricting the regression function $g\left(X\right)$ to be a function η of a linear combination of X, we arrive at the class of generalized linear regression models (GLM) with the mathematical representation.

[8] It is especially dangerous to bootstrap when the asymptotic distribution of a score statistic is unknown.

$$C\left(Y|X\right)=\eta\left(X\beta\right),$$

Where η is called the link function, examples for this class are the well-known linear and logistic regression models. The former is commonly expressed as

$$\mathbb{E}\left[Y|X\right]=X\beta$$

with the additional assumption that $Y\sim N\left(X\beta,\sigma^2 I\right)$. For the latter, let $Y\in\{0,1\}$ be a Bernoulli distributed random variable with $p\coloneqq\mathbb{P}\left(Y=1|X\right)$ and η the logit function, then we arrive at the familiar binary logistic regression model

$$\mathbb{E}\left[Y|\mathrm{X}\right]=\mathbb{P}(Y=1|X)=\mathrm{logit}\left(X\beta\right).$$

In the context of machine learning experiments, Y represents the evaluation metric $m(y,\hat{y})$ used to quantify classifier performance, and X represents the model or algorithm used to generate the predictions $\hat{y}$ but can also incorporate computed or annotated input characteristics. This brief and abstract discussion is sufficient to show that regression models allow detailed and fine-grained comparisons of the expected risks based on the evaluation data collected during a machine learning experiment. One key aspect differentiating LMEMs from conventional generalized linear models is their ability to easily model complex non-iid sampling data (more details in 2.2). This ability, along with a highly elaborated estimation and hypothesis testing theory [PB00] based on the maximum likelihood principle and the generalized likelihood ratio test (GLRT), make LMEMs the mean of choice to analyze the complex evaluation data obtained from machine learning experiments and redeem the machine learning researcher to resort to specialized and flawed procedures matched to different evaluation metric classes to deal with multiple predictions from different meta-parameter configurations and multiple datasets (see [Dro+20], Chapters 4 and 5, respectively).

I begin my exposition by briefly discussing hypothesis testing principles and shortcomings. This discussion includes less obvious assumptions behind (re-)sampling-based significance tests that can severely limit the scope of their applicability. Finally, I will present the mechanics and mathematics of the GLRT and showcase its combined application with LMEMs to reanalyze the evaluation data obtained during an interactive machine translation experiment conducted by [KBR20].

2.1 The Principles of Statistical Hypothesis Testing

The fundamental goal of statistical hypothesis testing is to decide between two mutually exclusive and exhaustive sets of hypotheses. One is called the null hypothesis H_0, and the other is called the alternative hypothesis H_1, by evidence obtained from observed random samples. Every statistical test, regardless if it is parametric, non-parametric, or sampling-based, starts by assuming the correctness of the null hypothesis and, based on this assumption, derives the distribution of a so-called test statistic[9] which is used to distinguish between H_0 and H_1. The crucial step is to derive the distribution of this statistic under the null hypothesis. If the observed value of the test statistic is very unlikely under H_0 — lower or equal than a predefined significance level $\alpha \in (0,1)$ — the null hypothesis is rejected in favor of the alternative hypothesis.

For parametric tests, it is sometimes possible to derive this distribution analytically[10] for finite sample sizes, based on the assumed data distribution and known parameters of this distribution. However, in most cases, the distribution of the test statistic can only be approximated via asymptotic (large sample) arguments.

Let us consider the hypothesis testing problem about the expected value of a distribution F with finite expectation and non-zero variance. The critical theorem that facilitates arriving at a useful distribution for a test about the mean is the Central Limit Theorem. The classical form can be stated in the following way[11]:

Theorem 2.1 (Classical Central Limit Theorem)**.**
Let $\bar{X}_N$ be the arithmetic mean of the first N of a sequence of independent and identically distributed scalar random variables $X_1, X_2, \ldots$. Let us further assume that $\mathbb{E}\left[Y_i^2\right] < \infty$ (meaning that the data are drawn from a distribution with finite expectation μ and variance σ^2), and let F_N denote the cumulative distribution function (CDF) of $\sqrt{N}\frac{\bar{X}_N-\mu}{\sigma}$, then

$$F_N(x) \xrightarrow{N\to\infty} \Phi(x), \qquad \forall x \in \mathbb{R},$$

where $\Phi(x)$ denotes the cdf of a standard Gaussian random variable. Note that the result also holds when σ is unknown but can be replaced by a consistent estimator.

[9] Following [LM12], we define as test statistic any function of the observed data whose numerical value dictates whether H_0 is accepted or rejected.

[10] If this is possible, the corresponding test is usually called *exact*.

[11] Formal derivations and proofs for several variants of the asymptotic argument can be found in [Vaa98], Chapter 2.

To conduct a hypothesis test about the test statistic of the mean, we use Theorem 2.1 to approximate the distribution of $\bar{X}_N$ by a Gaussian distribution. The correctness of this approximation increases as N increases. This statement about the approximate distribution of the mean of samples of size N can be given as follows:

$$\bar{X}_N \overset{app}{\sim} \mathcal{N}(\mu, \frac{\sigma^2}{N}). \tag{2.1}$$

It is important to stress that the approximate normal distribution of the mean $\bar{X}_N$ as stated in (2.1) follows from Theorem 2.1, irrespective of the shape of the distribution from which the samples $X_1, X_2, \ldots$ are drawn.

Let us assume that we know that our data were drawn from a distribution with standard deviation σ and that we want to test if the mean μ of this distribution equals μ_0 or not. Then, the null hypothesis reads.

$$\mathrm{H}_0 : \mu - \mu_0 = 0,$$

and the alternative hypothesis is

$$\mathrm{H}_1 : \mu - \mu_0 \neq 0.$$

For concreteness, let us use an example from [Coh95] where we know $\sigma = 50$, and we want to test if the expected value of the data generating distribution is $\mu_0 = 25$. To test this hypothesis, we sample 100 observations from which we estimate a mean $\bar{x}$[12] of 15, yielding a Z-score[13] of

$$Z = \sqrt{N}\frac{\bar{x} - \mu_0}{\sigma} = \sqrt{100}\frac{15 - 25}{50} = -2.$$

Let us further assume that we want to control the Type I error[14] at $\alpha = .05$[15]. Figure 2.1

[12] In this case, $\bar{x}$ serves as an estimator for μ. The law of large numbers justifies this usage. Some authors use the symbol $\hat{\mu}_N$ instead of $\bar{x}_N$ to stress this point.

[13] According to our definition of test statistic, following [LM12], both $\bar{x}$ and Z qualify as test statistics.

[14] Type I error means that we decide to reject the null hypothesis based on our test, but the null hypothesis is the correct model.

[15] This means that given the null hypothesis is correct, we want to set the probability that our test makes a Type I error at 5%. If the null hypothesis contains more than one alternative, then α bounds the supremum of the probability that our test makes a Type I error of more than 5%. The ability

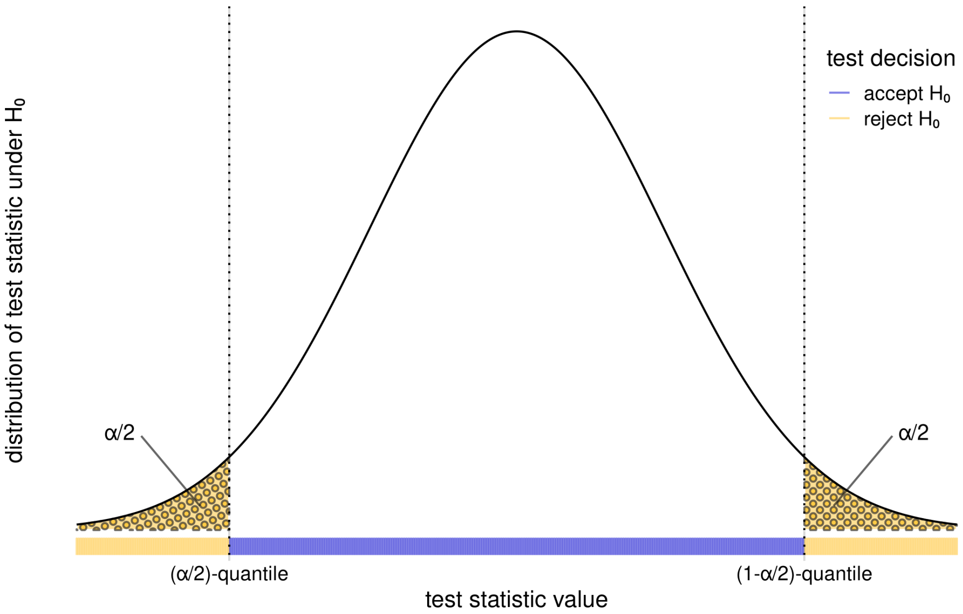

Figure 2.1: Critical region of two-tailed Z-test.

shows the shape of a standard normal distribution, which is the approximate distribution of our test static under the null hypothesis. Based on the nature of our hypothesis pair and our choice of α, we can partition the range of our test statistic into two regions. One is called the acceptance region, which comprises all observable test statistic values deemed compatible with the null hypothesis. The second region is called the rejection region. This region is the set of all the test statistic's observable values deemed incompatible with the null hypothesis. When we observe a value in this set, we reject the null hypothesis in favor of the alternative. The rejection region is constituted by the distribution's tails, which for our test is $(-\infty, -1.96] \cup [1.96, \infty)$. In our case, the observed value $Z = -2$ is in the rejection region, so we know that obtaining this result by chance under H_0 is less than 5%. Thus, we reject H_0 at an $\alpha = 0.05$ level and call the difference between $\bar{x}$ and μ_0 statistically significant.

A hypothesis test like the previous is called a two-sided test because the alternative hypothesis encompasses both possibilities $\mu < \mu_0$ and $\mu > \mu_0$. If it only contains one of these, the corresponding test is one-sided. Let us stay in the setting of the previous example, but now we are interested in testing whether μ is less than μ_0. The corresponding hypotheses pair reads:

to control the Type I error probability at a nominal rate is one of the most essential properties of a statistical significance test.

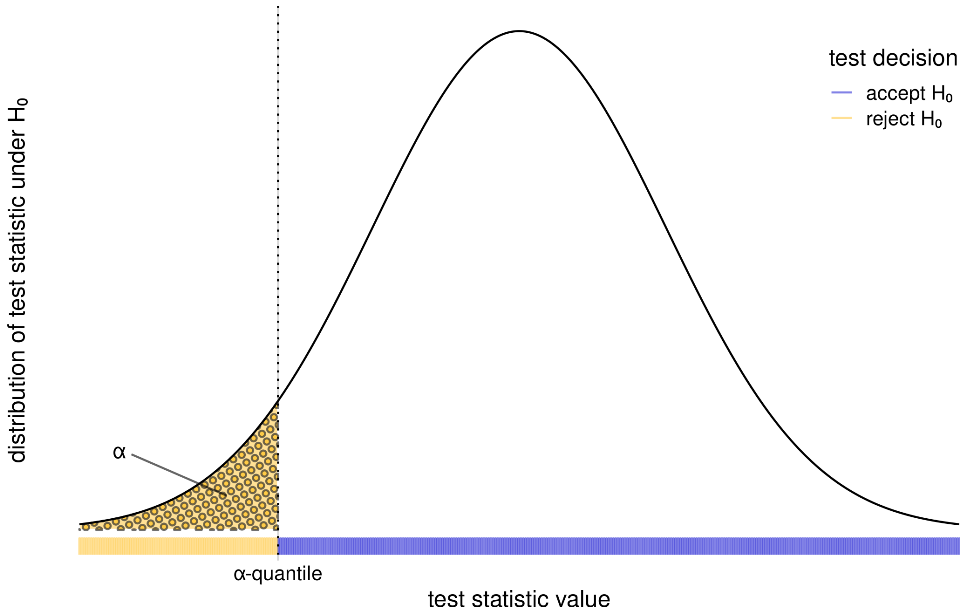

Figure 2.2: Critical region of left-tailed Z-test.

$$
\begin{aligned}
\mathrm{H}_0&: \quad \mu \geq \mu_0 \\
\mathrm{H}_1&: \quad \mu < \mu_0
\end{aligned}
$$

The test statistic is identical to the two-sided test, but the rejection region differs. As shown in Figure 2.2, we can put the total mass of α in the left tail instead of splitting it for a two-sided test. Thus our rejection region now is $(-\infty, -1.64]$. Again, the observed value $Z = -2$ is in the rejection region, and we, therefore, decide to reject the null hypothesis and assume the alternative to be correct.

Discussion. The reasoning behind the z-test is similar to any parametric significance test. The pivotal problem is to derive the sampling distribution for the given test statistic. In the case of sum-based test statistics and large enough sample sizes, the Central Limit Theorem can be applied to approximate the sampling distribution by a normal distribution. Thus, for NLP and data science applications where the standard evaluation metric is based on a mean of sample evaluations, the family of approximate z-tests allows us to test the statistical significance of result differences between performance evaluation of machine learning models.

One problem of applying the classical Central Limit Theorem to NLP applications is the assumption of independence of the samples for which the test statistic is calculated. This assumption is often violated in NLP if test sets consist of sentences of the same document.[16]

Another problem is the precise understanding of the phrase "mean of samples" that defines the test statistic in the Central Limit Theorem. This term applies to test statistics in NLP and data science that are calculated as means of evaluation scores that have been computed separately for each sentence in a test set. Examples are accuracy scores or the TER score [Sno+06] that we used to evaluate machine translation systems in Chapter 3. Generally, any evaluation score computed as an average over sentence-level evaluation scores qualifies as a "mean" test statistic to which the Central Limit Theorem applies. Thus, no matter the distribution of the sentence-wise evaluation score, a significance test like the z-test will be applicable to the test statistic of the mean of sentence-wise scores over the test set.

The story is different for corpus-wise evaluation measures such as BLEU [Pap+02] that are computed on a corpus level, i.e., by accumulating all statistics for n-gram precision and brevity penalty over the whole test set and then combining these statistics in a nonlinear way. Similarly, corpus-level versions of precision, recall, or F1-score, where statistics on true positives, false positives, and false negatives are accumulated over the test items and then pro-rated, are nonlinear combinations of test statistics. Thus, even if the elementary statistics of n-gram counts, true positives, or false positives are normally distributed because they are sums over test set items, their non-mean-like aggregation at the corpus level does not follow a normal distribution in general and has to be established for the specifc case.

To summarize, since many standard performance evaluation measures in NLP and data science do not qualify as "mean of samples," techniques for statistical significance testing will be needed that can establish sampling distributions for test statistics without explicit reference to the Central Limit Theorem.[17] We will describe two such hypothesis tests in the next section.

[16] The problem of clustered test samples is to be distinguished for another type of independence violation discussed in [Yeh00]. These concern positive correlations between pairs of systems, e.g., a baseline and a refined system, for which the significance of result differences is to be assessed. [Yeh00] suggests tests for matched pairs as a remedy. The model-based significance test discussed in the following can be seen as an instance of a matched-pair test.

[17] The problem of corpus-level measures versus sentence-level measures often leads to confusion in attempts to match evaluation metrics to significance tests. For example, in the matching table of Chapter 3 of [Dro+20], only sentence-level test statistics can be matched to parametric tests like the t-test. However, this assumption is not made explicit.

2.1.1 (Re-)Sampling-Based Significance Tests I: Bootstrapping

Bootstrap resampling has become a prevalent technique for statistical significance testing in NLP and data science due to its ability to construct sampling distributions for virtually any test statistic, without knowing its actual sampling distribution and without making assumptions about the parametric distribution of the population. It has been developed in biostatistics [ET93] and has quickly been adopted in the machine learning community [HTF08]. In the following, we will restrict our attention to nonparametric bootstrap resampling and refer to this technique with the shorthand "the bootstrap".

The principle of the nonparametric bootstrap is to replace the unknown CDF of the data-generating process with an empirical estimator based on the sample. And then, use this estimator to draw bootstrap samples to generate a sampling distribution of the test statistic. Repeated sampling (with replacement) from the sample itself is a computationally efficient implementation of this idea. In the following, we will consider bootstrap methods for the test statistic of the difference in corpus-level performance evaluation scores $S_A - S_B$ on a test set for machine learning models $\mathcal{A}$ and $\mathcal{B}$. The null hypothesis is that the scores of systems $\mathcal{A}$ and $\mathcal{B}$ are random samples from the same distribution. First, the actual test statistic is computed on the test data. Next, the sample mean of the test statistic is calculated on the bootstrapped data, i.e., the test statistic is computed on bootstrap samples of equal size to the test set and averaged over bootstrap samples. To compute the sampling distribution of the test statistic under the null hypothesis, we employ the “shift” method described in [Nor89]. Here it is assumed that the sampling distribution of the null hypothesis and the bootstrap sampling distribution have the same shape but a different location. The location of the bootstrap sampling distribution is shifted so that it is centered according to the null hypothesis. This is achieved by subtracting the expected value of the score difference, estimated by the sample mean of the test statistic on the bootstrap samples, from each value. Then, a p-value is computed directly from the percentage of trials where the (shifted) test statistic is greater than or equal to the actual statistic. Thus we directly calculate the probability of obtaining a sample result under the null hypothesis as extreme or more extreme than the score difference observed on the original test set. Following standard practice in statistical significance testing, assessing statistical significance at a given α level is common if the p-value is less than or equal to α. However, it is considered good practice to report p-values directly and treat them as the smallest α at which statistical significance can be assessed [McS+19].

Pseudocode sketching of a two-sided bootstrap significance test for evaluation score differences is given below:

Algorithm 2.1 (Bootstrap Test)

```
Given test set outputs $(A_0, B_0) = (a_i, b_i)_{i=1}^N$, where $a_i$ is the output
of system $\mathcal{A}$, and $b_i$ is the output of system $\mathcal{B}$, on test instance $i$.

Compute score difference $\Delta S_0 = S(A_0) - S(B_0)$ on test data.

For $k = 1, \dots, K$:
    Generate bootstrap dataset $S_k = (A_k, B_k)$ by sampling $N$ examples
    from $(a_i, b_i)_{i=1}^N$ with replacement.
    Compute score difference $\Delta S_k = S(A_k) - S(B_k)$ on bootstrap data.
Compute $\overline{\Delta S_k} = \frac{1}{K} \sum_{k=1}^K \Delta S_k$.
Set $c = 0$.
For $k = 1, \dots, K$:
    If $|\Delta S_k - \overline{\Delta S_k}| \geq |\Delta S_0|$
      $c++$

$p = c/K$.

Reject null hypothesis if $p$ is less than or equal to specified rejection level $\alpha$.
```

Discussion. The key assumption of the bootstrap can be described formally with [Can+06] as the *bootstrap substitution principle.* This principle states that an approximation of a probability distribution of the quantity $u(Y, F)$, where $Y = (Y_1, Y_2, \dots, Y_N)$ is randomly sampled from F, can be constructed by replacing F by a resampling model $\hat{F}$ from which samples Y^* are drawn such that.

$$P\{u(Y, F) \leq u|F\} \approx P\{u(Y^*, \hat{F}) \leq u|\hat{F}\}. \tag{2.2}$$

A standard nonparametric resampling model is the empirical distribution function $\tilde{F}$, which estimates the distribution F by assigning probability $1/N$ to each sample $Y_i, i = 1, \dots, N$. The sample's representativeness fundamental to the bootstrap is measured by the size of the approximation error in the bootstrap substitution principle as the sample size N goes to infinity called *bootstrap consistency* in [Can+06]. Bootstrap methods can be inconsistent if the left-hand side and the right-hand side of the bootstrap substitution equation (2.2) do not converge to the same value, no matter how large the sample size is. [BBK12] have shown that p-values computed on bootstrap samples from one test set may not indicate actual result differences on another test set if there is a significant domain shift between both test sets. This extreme inconsistency is a problem for any inference procedure. However, bootstrap inconsistency can result from complex interactions of resampling schemes, test statistics, and data distributions. [Can+06] describes diagnostics for various bootstrap inconsistencies usually ignored in NLP and data science applications.

In contrast to our goal of incorporating randomness due to meta-parameters or test data into significance testing, bootstrap tests are usually applied to a single test set on which a pair of selected systems is to be compared. [Sel+21] presented a so-called "multi-bootstrap" technique that resamples both from random seeds and test set instances to estimate the significance of the result difference between the average performance evaluation scores of two systems. In this setup, the more powerful and, thus, the preferred paired design is only possible if random seeds are identical for compared systems, e.g., in fine-tuning setups. The unpaired design is more flexible. However, it suffers the usual loss in power since it has to assume zero covariance between the performance evaluation scores of the compared systems.[18]

2.1.2 (Re-)Sampling-Based Significance Tests II: Permutation

The permutation test, known as the (approximate) randomization test, dates back to [Fis35]. Similar to the bootstrap test, it is based on random sampling. However, it does not assume the representativeness of the test sample, which can be problematic in NLP data. Instead, it directly tests the weak assumption that two machine learning systems are related without, in fact, assuming the population distribution of the evaluation scores either.

The null hypothesis of the permutation test is that systems $\mathcal{A}$ and $\mathcal{B}$ are identical. Thus, under the null hypothesis, outputs for the same input are exchangeable, i.e., any output produced by one of the systems on a test sentence could have been created just as likely by the other system. So shuffling the sentence-wise outputs between the two systems with equal probability, and recomputing the test statistic, allows approximating a p-value by computing the percentage of trials where the test statistic computed on the shuffled data is greater than or equal to the test statistic calculated on the test data.

For a test set of N sentences, there are 2^N different ways to shuffle the sentence-wise

[18] See, for example, the discussion of the two-sample t-test versus the paired sample t-test in [Coh95].

outputs between the two systems. If all permutations are considered, the randomization test is exact. Approximate randomization produces a subset of all possible shuffles; however, the more shuffles are evaluated, the better the approximation of the p-value. Again, it is considered good practice to report p-values directly instead of just assessing statistical significance at a given α-level [McS+19].

A sketch of an algorithm for a two-sided approximate randomization test for the significance of performance score differences is given below:

Algorithm 2.2 (Permutation Test)

```
Given test set outputs $(A_0, B_0) = (a_i, b_i)_{i=1}^N$, where the first element
in the ordered pair $(a_i, b_i)$ is the output of system $\mathcal{A}$, and the second
element is the output of system $\mathcal{B}$, on test instance $i$.

Compute score difference $\Delta S_0 = S(A_0) - S(B_0)$ on test data.

Set $c = 0$.

For $r = 1, \ldots, R$:
    Compute shuffled outputs $(A_r, B_r)$ where for each $i = 1, \ldots, N$:
    $\mathrm{swap}(a_i, b_i) = \begin{cases} (a_i, b_i) & \text{with probability } 0.5, \\ (b_i, a_i) & \text{with probability } 0.5. \end{cases}$
    Compute score difference $\Delta S_r = S(A_r) - S(B_r)$ on shuffled data.
    If $|\Delta S_r| \geq |\Delta S_0|$
      $c++$

$p = c/R$.

Reject null hypothesis if $p$ is less than or equal to specified rejection level $\alpha$.
```

Discussion. The permutation test rests on the simple and powerful principle of *stratified shuffling*[19] [Nor89], which allows generating the sampling distribution by shuffling outputs between the two systems within blocking strata [20]. Based on this principle, the inventors

[19] The crucial assumption behind any permutation test is that the observations are exchangeable under the null hypothesis.

[20] Along randomization and replication, *blocking* is on of the most elementary principles of experimental design [Mon17]. It encompasses all design techniques aiming to improve the precision of fixed effects comparisons by eliminating the variability transmitted form nuisance factors; that is factors that influence the experimental response but in which the experimenter is not directly interested.

of the bootstrap rate the permutation test as follows:

> When there *is* something to permute, [...] it is a good idea to do so, even if other methods like the bootstrap are also brought to bear. [ET93]

This statement showcases both the advantages and disadvantages of the permutation test. Strata for shuffling outputs between the two systems must be identified to generate null-hypothesis conditions. Strata are given naturally in NLP test sets where each sentence corresponds to a stratum. Outputs can be sentence-wise evaluation scores or count statistics that are accumulated over the whole test corpus, for example, sentence-level TER [Sno+06] or sentence-level n-gram counts in BLEU [Pap+02], respectively. Suppose the goal is to compare two machine learning systems on the same sentences of a test set. In that case, a permutation test is easily implemented, allowing a powerful (i.e., high probability of rejecting H_0 when it is false) assessment of statistical significance. The latter has been shown formally by comparing permutation and parametric tests for large samples [Hoe52].

To sum up, the permutation test is the method of choice if the only goal is to assess the statistical significance of a difference in evaluation scores between two instances on the same test set. However, to apply a permutation test described above, a meta-parameter configuration for each algorithm and a test set must be fixed. An extension of the permutation principle to a k-sample test that incorporates blocking –and thus facilitating the simultaneous analysis of several algorithm instances– can be constructed based on the theory presented by [SW99], which is implemented in [Hot+08]. Independent of this work [Cla+11] suggested an ad-hoc procedure that permutes within a block without any theoretical justification. These methods are appropriate for an overall comparison but cannot facilitate a fine-grained analysis considering data properties.

A more flexible framework for statistical significance testing that allows multiple comparisons without increased Type I error, and enables elegant incorporation of variability due to optimization and test data, is the model-based approach to significance testing. We will describe this technique in section (2.2).

Typical comparison techniques based on blocking are the paired t-test [Mon17] or the approximate randomization test [RM05].

2.1.3 Excursion: Multiple Comparisons and Elementary Post-hoc Procedures

The theory of multiple testing, or multiple comparisons as it is sometimes called, is concerned with the problem of testing $m > 1$ hypotheses in a sample simultaneously. To formalize the problem let us consider the situation where m tests are performed for a given sample X with corresponding pairs of null hypothesis $\mathrm{H}_0^{(j)}$ and alternative hypothesis $\mathrm{H}_A^{(j)}$ for $j = 1, \ldots, m$. Let $m_0 \leq m$ denote the number of tests for which the null hypothesis is correct. Further, let the test decisions for each of the m tests be based on the corresponding test statistic $T_1(X), T_2(X), \ldots, T_j(X), \ldots, T_m(X)$. Statistical tests are constructed so that the null and the alternative hypothesis are mutually exclusive. Consequently, a statistical test can yield one of the following four possible results:

1. H_0 is true and the test accepts H_0.

2. H_0 is true but the test rejects H_0 (this is called a Type I error).

3. H_A is true but the test accepts H_0 (this is called a Type II error).

4. H_A is true and the test rejects H_0.

Table 2.1 provides the standard notation to summarize the outcome of m test results. In this table, V denotes the number of Type I errors that have happened, and T is the number of Type II errors. Naturally, we would like both numbers to be as small as possible. Still, unfortunately, this optimal situation is not achievable with a finite sample size because the probability of a Type I error and the probability of a Type II error are antagonistically related for a statistical test. In concreto, a low Type I error probability necessarily leads to an increased Type II error probability and vice versa. Usually, the Type I error probability is fixed at a certain level $\alpha \in (0, 1)$ (named the α-level of the test), and one chooses or constructs a test for a specific situation such that the Type II error probability is as small as possible, or equivalently that the power of the test – defined as the probability to reject the null hypothesis when the alternative is true – is maximized. The essential property of a statistical test is its ability to control the Type I error probability at α, which means that $\mathbb{P}(\textit{reject}\, \mathrm{H}_0) \leq \alpha$ where $\mathbb{P}$ denotes the probability measure under H_0. An important question is how to generalize this property, the ability to control the probability of a certain misjudgment, to the multiple testing situation.

	H_0 accepted	H_0 rejected	Total
H_0 is true	U	V	m_0
H_A is true	T	S	$m - m_0$
Total	$m - R$	R	m

Table 2.1: Notation for multiple testing

Many measures have been suggested for this purpose [HT09]. Common generalizations are the *per-family error rate* $PFER := \mathbb{E}[V]$, the *per-comparison error rate* $PCER := \frac{\mathbb{E}[V]}{m}$ and most important the *family wise error rate* $FWER := \mathbb{P}(V > 0)$ which is the probability to make at least one Type I error within the family of m tests. Unfortunately, for very large m, controlling the FWER at an acceptable level leads to procedures with deficient power to detect actual signals in the data, implying that such a procedure will miss a lot of true signals and therefore have a high probability of generating Type II errors.

There exists a wide variety of procedures that guarantee control of FWER in different multiple testing situations. I limit my exposition to the most popular method, namely the *Bonferroni correction* [Bon35] and one of its variants, the *Bonferroni-Holm step-up* procedure [Hol79].

Let us first define the event $B_j := \left\{ \mathrm{H}_0^{(j)} \textit{ rejected} \right\}$. For the corresponding test statistic T_j and an associated critical value T_{crit,α_j} an equivalent characterization is $B_j = \left\{ T_j \geq T_{crit,\alpha_j} \right\}$. By definition, the critical value for a test statistic is chosen such that $\mathbb{P}(B_j) = \alpha_j$ where α_j is the α-level of the jth-test. Without loss of generality, we assume that the hypotheses are ordered so that the null hypothesis is true for the first m_0 hypotheses. Thus, we can write.

$$FWER = \mathbb{P}(V > 0) = \mathbb{P}\left(\bigcup_{j=1}^{m_0} B_j \right).$$

It follows from the sub-additivity of the probability measure that.

$$FWER \leq \sum_{j=1}^{m_0} \mathbb{P}(B_j) = \sum_{j=1}^{m_0} \alpha_j \leq \sum_{j=1}^{m} \alpha_j.$$

Therefore, whenever $\sum_{j=1}^{m} \alpha_j$ is bounded by some $\alpha \in (0,1)$ then the FWER for the m simultaneous tests is also bounded by α. The Bonferroni procedure exploits this fact. The Bonferroni correction suggests to choose the individual $\alpha_j \in (0,1)$ such that $\sum_{j=1}^{m} \alpha_j = \alpha$

for some predefined $\alpha \in (0,1)$ and reject $\mathrm{H}_0^{(j)}$ whenever $p_j := \mathbb{P}\left(T_j \geq t_{j,obs}\right) \leq \alpha_j$ where $t_{j,obs}$ is the observed test statistic for test j. The standard choice is $\alpha_j = \frac{\alpha}{m}$.

We have seen that the argument behind the Bonferroni correction makes no use of the actual distribution of $(T_1, T_2, \ldots, T_m)$. Therefore this procedure has the favorable property that guarantees FWER control for all possible joint distributions of the test statistics. Still, on the other hand, the actual FWER may be much smaller than the nominal α. For instance, when $m_0 \ll m$ or the test statistics are positively correlated. A multiple testing procedure with this property is called *conservative*. This property is typically associated with reduced power to detect valid signals in the data.

An improvement (in terms of a power gain) of the Bonferroni correction is the Bonferroni-Holm procedure, which results from applying the *closed testing principle* [MEG76]. Let $p_{[1]} \leq p_{[2]} \leq \cdots \leq p_{[j]} \leq \cdots \leq p_{[m]}$ be the ordered sequence of p-values obtained by the m individual tests and $\mathrm{H}_0^{[j]}$ the corresponding sequence of null hypotheses. Let k be the smallest index j such that $p > \frac{\alpha}{m+1-j}$. Then the Bonferroni-Holm procedure rejects all m hypotheses if no such k exists and rejects all $\mathrm{H}_0^{[j]}$ with $j < k$ otherwise.

In contrast to the Bonferroni adjustment, the acceptance or rejection of a particular hypothesis $\mathrm{H}_0^{(j)}$ depends on the value of all other test statistics T_i for $i \neq j$. The benefit of this is an enlarged rejection region and, thus, an increased power compared to the Bonferroni adjustment.

The Bonferroni-Holm procedure is an example of a so-called *step-down procedure*. Stepwise procedures are characterized by making test decisions based on an ordered sequence of p-values. Step-down procedures start from the smallest one, each time checking if a condition is satisfied and stopping the first time this condition is met. Then all null hypotheses with a smaller index than the stopping index are rejected.

2.2 Model-Based Algorithm Comparison: Toolbox

The central concept of model-based significance testing is to express the hypotheses under investigation by the parameters of the evaluation data probability distribution. We will use LMEMs to describe this distribution and the test-data-based parameter estimates to conduct inference. The test of choice in this paradigm is the generalized likelihood ratio test dating back to principles formulated by [NP33]. We will follow the exposition in [Vaa98].

Linear Mixed Effects Models: General Form of Model

A linear mixed effects model (LMEM) is an extension of a standard linear model that allows a rich linear structure in the random component of the model, where effects other than those that can be observed exhaustively (so-called *fixed effects*) are treated as a random sample from a larger population of normally distributed random variables (so-called *random effects*).

Given a dataset of N input-output pairs $\{(\mathbf{x}^n, y^n)\}_{n=1}^N$, the general form of an LMEM is

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\epsilon},$$

Where $\mathbf{X}$ is an $(N \times k)$-matrix and $\mathbf{Z}$ is an $(N \times m)$-matrix, called model- or design-matrices (both are known), which relate the unobserved vectors $\boldsymbol{\beta}$ and $\mathbf{b}$ to $\mathbf{Y}$. $\boldsymbol{\beta}$ is a k-vector of fixed effects and $\mathbf{b}$ is an m-dimensional random vector called the random effects vector. $\boldsymbol{\epsilon}$ is an N-dimensional vector called the error component. The random vectors are assumed to have the following distributions:

$$\mathbf{b} \sim \mathcal{N}(0, \psi_\theta),$$
$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Lambda}_\theta),$$

where ψ_θ and $\boldsymbol{\Lambda}_\theta$ are covariance matrices parameterized by the vector θ.

The definition of an LMEM implies a definition of the distribution of the data vector $\mathbf{Y}$. In the context of the LMEM theory, we consider three important distributions, the first of which is the distribution of $\mathbf{Y}|\mathbf{b}$. When we fix $\mathbf{b}$, the only random component left is $\boldsymbol{\epsilon}$. Thus the conditional distribution of $\mathbf{Y}$ given $\mathbf{b}$ is

$$\mathbf{Y}|\mathbf{b} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b}, \boldsymbol{\Lambda}_\theta).$$

This distribution is the basis for deriving the so-called mixed model equations or Henderson equations which provide estimators for the unknown quantities $\boldsymbol{\beta}$ and $\mathbf{b}$.

The second distribution of importance is the unconditional distribution of $\mathbf{Y}$. We defined $\mathbf{Y}$ as a linear mapping of the independent zero-mean Gaussian variables $\mathbf{b}$ and $\boldsymbol{\epsilon}$.

Thus $\mathbf{Y}$ is also a Gaussian with expected value $\mathbf{X}\boldsymbol{\beta}$. Since the variance [21] can be written as $\mathbb{V}(\mathbf{Z}\mathbf{b}) = \mathbf{Z}\boldsymbol{\psi}_\theta\mathbf{Z}^\top$, we get

$$\mathbf{Y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \mathbf{Z}\boldsymbol{\psi}_\theta\mathbf{Z}^\top + \boldsymbol{\Lambda}_\theta).$$

Note that $\mathbf{b}$ doesn't occur in this distribution. Instead, random effects enter the distribution only via the covariance matrix $\mathbf{Z}\boldsymbol{\psi}_\theta\mathbf{Z}^\top$. This fact reveals one of the main usages of mixed models: the convenient modeling of complex covariance structures when the data were not generated in the usual i.i.d. sampling fashion.

We derive the joint distribution of $\mathbf{b}$ and $\mathbf{Y}$ to complete our enumeration of important distributions. For this purpose, we stack $\mathbf{b}$ and $\mathbf{Y}$ together in a vector. Because both variables are multivariate Gaussians, the resulting vector is also a multivariate Gaussian, where the expected values and the block diagonal parts of the covariance matrix are inherited from $\mathbf{b}$ and $\mathbf{Y}$. Note that the covariance [22] of $\mathbf{b}$ and $\mathbf{Y}$ is $\text{Cov}(\mathbf{b}, \mathbf{Y}) = \psi_\theta\mathbf{Z}^\top$ and let $\mathbf{V} = \mathbf{Z}\boldsymbol{\psi}_\theta\mathbf{Z}^\top + \boldsymbol{\Lambda}_\theta$, then

$$\begin{bmatrix}\mathbf{b}\\ \mathbf{Y}\end{bmatrix} \sim \mathcal{N}(\begin{bmatrix}\mathbf{0}\\ \mathbf{X}\beta\end{bmatrix}, \begin{bmatrix}\psi_\theta & \psi_\theta\mathbf{Z}^\top\\ \mathbf{Z}\psi_\theta & \mathbf{V}\end{bmatrix}).$$

Finally, let us say some words on the usage of LMEMs. As mentioned, the most common application of LMEMs is to model complex covariance structures in the data when the usual i.i.d. assumptions fail to apply. Typical use cases are repeated or grouped, and thus non-independent, measurements. In this case, LMEMs provide a neat means to provide correct statistical inference about fixed effects (usually of primary interest to the analyst). Like other linear models, they can also predict outcomes when the covariates are known. This prediction can be based on the unconditional distribution of $\mathbf{Y}$ or, when random effects are known, on the conditional distribution $\mathbf{Y}|\mathbf{b}$. Predictions based on the latter are usually associated with a smaller prediction uncertainty (via different covariance matrices). Furthermore, like Bayesian or other generative models, LMEMs can generate synthetic data. A special case of LMEMs are models where $\mathbf{X} = \mathbf{0}$ and which

[21] $\mathbb{V}(\mathbf{Y}) = \mathbb{E}((\mathbf{Y} - \mathbb{E}(\mathbf{Y}))(\mathbf{Y} - \mathbb{E}(\mathbf{Y}))^\top) = \mathbb{E}((\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\epsilon} - \mathbf{X}\boldsymbol{\beta})(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\epsilon} - \mathbf{X}\boldsymbol{\beta})^\top) = \mathbb{E}((\mathbf{Z}\mathbf{b} + \boldsymbol{\epsilon})(\mathbf{b}^\top\mathbf{Z}^\top + \boldsymbol{\epsilon}^\top)) = \mathbb{E}(\mathbf{Z}\mathbf{b}\mathbf{b}^\top\mathbf{Z}^\top + \boldsymbol{\epsilon}\mathbf{b}^\top\mathbf{Z}^\top + \boldsymbol{\epsilon}\boldsymbol{\epsilon}^T + \mathbf{Z}\mathbf{b}\boldsymbol{\epsilon}^\top) = \mathbf{Z}\mathbb{E}(\mathbf{b}\mathbf{b}^\top)\mathbf{Z}^\top + \mathbb{E}(\boldsymbol{\epsilon})\mathbb{E}(\mathbf{b}^\top)\mathbf{Z}^\top + \mathbb{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T) + \mathbf{Z}\mathbb{E}(\mathbf{b})\mathbb{E}(\boldsymbol{\epsilon}^\top) = \mathbf{Z}\ \mathbb{V}(\mathbf{b})\mathbf{Z}^\top + \mathbb{V}(\boldsymbol{\epsilon}) = \mathbf{Z}\boldsymbol{\psi}_\theta\mathbf{Z}^\top + \boldsymbol{\Lambda}_\theta$.

[22] $\text{Cov}(\mathbf{b}, \mathbf{Y}) = \mathbb{E}((\mathbf{b} - \mathbb{E}(\mathbf{b}))(\mathbf{Y} - \mathbb{E}(\mathbf{Y}))^\top) = \mathbb{E}(\mathbf{b}(\mathbf{Z}\mathbf{b} + \boldsymbol{\epsilon})^\top) = \mathbb{E}(\mathbf{b}(\mathbf{b}^\top\mathbf{Z}^\top + \boldsymbol{\epsilon}^\top))$
$= \mathbb{E}(\mathbf{b}\mathbf{b}^\top\mathbf{Z}^\top + \mathbf{b}\boldsymbol{\epsilon}^\top) = \mathbb{E}(\mathbf{b}\mathbf{b}^\top)\mathbf{Z}^\top + \mathbb{E}(\mathbf{b})\mathbb{E}(\boldsymbol{\epsilon}^\top) = \mathbb{V}(\mathbf{b})\mathbf{Z}^\top = \psi_\theta\mathbf{Z}^\top$.

therefore do not contain fixed effects. These models are called random effect models or variance component models. Their purpose is to partition the total observed variance of the outcome according to different sources. One application of these models is reliability studies.

2.2.1 Linear Mixed Effects Models: Specification Example

Let us illustrate an actual LMEM specification for analyzing the data in the hypothetical "lexical decision" experiment of [Bar+13]. The resulting model is far more complex than the models needed for algorithm comparison. Nevertheless, this example is illustrative and will show the inclined reader how to specify complex LMEMs in detail. In this experiment, four strings of characters were presented to four human subjects who had to decide whether or not the string formed an English word. The time from stimulus presentation to subject response (henceforth response time) was measured. The strings belong to two categories which are assumed to impact the response time. For simplicity, we assume that strings 1 and 2 belong to Category A and 3 and 4 to Category B. The experiment was carried out to test this assumption.

To analyze these data, one has to build a statistical model incorporating the variables of interest. The most basic model we could start with is

$$y_{si} = \beta_0 + \beta_1 x_i + \epsilon_{si},$$

where y_{si} denotes the response time of subject s for character string i, x_i encodes the category of character string i (where 0 represents category A and 1 category B), and $\epsilon_{si} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_{error})$ is a random error component. The parameter β_0 is called the intercept. A simple calculation shows that $\beta_0 = \mathbb{E}[Y|x_i = 0]$ is the expected response time for items of category A. The parameter β_1 is called slope, and again a similar calculation shows that $\beta_1 = \mathbb{E}[Y|x_i = 1] - \mathbb{E}[Y|x_i = 0]$. It represents a measure of the difference of the expected response time for strings of category B versus strings of category A and thus is the main quantity of interest for the analysis of this experiment.

As Barr points out, this model can not be a correct representation of the data-generating mechanism of our experiment. A careful reading of the model definition reveals that we have assumed an error component independent of the measurements, implying that the observations y_i are independent. The experimental setup violates this implication because we take repeated measures from subjects and strings. Furthermore, the actual subjects and strings used in our experiment are just samples from larger populations, and we are not interested in obtaining a fixed effect-like estimate for the expected response time of subject s or item i (nor is it possible to do so with the data collected in this experiment). But we can account for the repeated measurements by incorporating appropriate random effects to model the covariance between measures. To specify the structure of the random effects, Barr argues that it is reasonable to assume individual differences exist between subjects when processing strings from categories A and B and that these differences can be different for both categories. He also argues that some strings can be processed faster than others. Therefore, he proposes the following model.

$$y_{si} = \beta_0 + b_s^{subject} + b_i^{item} + (\beta_1 + b_s^{slope})x_i + \epsilon_{si},$$

Where the symbols used in the simpler model retain their meaning, and $b_s^{subject}$ is a random variable that represents the distinctive deviation of subject s from β_0 (the overall expected response time for strings of category B), and b_i^{item} represent item specific deviations from this expectation. Therefore, these two random variables modify the model's intercept and are called random intercepts in the mixed models literature. The random variable b_s^{slope} is the subject-specific deviation from the global slope β_1 and is called a random slope. All the b are random variables, so we must specify a distribution. Let $\mathbf{b}_{si} := (b_s^{subject}, b_s^{slope}, b_i^{item})^\top$, then following Barr, we define

$$\mathbf{b}_{si} \overset{iid}{\sim} \mathcal{N}(\mathbf{0}, \begin{bmatrix} \sigma^2_{subject} & \sigma_{subject,\,slope} & 0 \\ \sigma_{subject,\,slope} & \sigma^2_{slope} & 0 \\ 0 & 0 & \sigma^2_{item} \end{bmatrix}),$$

Where $\sigma^2_{subject}$, σ^2_{slope}, and σ^2_{item} are the respective variances of the random variables, and $\sigma_{subject,\,slope}$ denotes the covariance of the two random effects for each subject.

Let us proceed to write the complete model for the experimental data so that we can see how $\mathbf{X}$, $\mathbf{Z}$, $\mathbf{\Lambda}_\theta$, ψ_θ and θ look like. Let us start by stacking the four model equations for a subject s together.

$$\underbrace{\begin{bmatrix} y_{s1}\\ y_{s2}\\ y_{s3}\\ y_{s4}\end{bmatrix}}_{\mathbf{y}_s} = \underbrace{\begin{bmatrix}1&0\\1&0\\1&1\\1&1\end{bmatrix}}_{\mathbf{F}} \begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix} + \begin{bmatrix} \underbrace{\begin{matrix}1&0\\1&0\\1&1\\1&1\end{matrix}}_{\mathbf{S}} & \underbrace{\begin{matrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{matrix}}_{\mathbf{I}}\end{bmatrix} \begin{bmatrix} b_s^{subject}\\ b_s^{slope}\\ \hdashline b_1^{item}\\ b_2^{item}\\ b_3^{item}\\ b_4^{item}\end{bmatrix} + \underbrace{\begin{bmatrix}\epsilon_{s1}\\ \epsilon_{s2}\\ \epsilon_{s3}\\ \epsilon_{s4}\end{bmatrix}}_{\epsilon_s} .$$

The matrix **F** encodes the presence or absence of the fixed effects β_0 and β_1 in the equations. Its first column corresponds to the intercept, which is present in all four equations, and thus it contains only 1s. The second column is associated with the slope, which is only present when the items belong to category B. The second term on the right-hand side represents the random effects. The first two random effects $b_s^{subject}$ and b_s^{slope} are subject-specific, and their presence in the model equations is given by **S**. Recall that $b_s^{subject}$ is a random intercept and b_s^{slope} a random slope, specific for subject s. The second block of random effects concerns the items. Each equation belongs to one item. Thus each has a unique item-specific random intercept which is ensured by the diagonal matrix **I**.

We have to put the four blocks for each subject together for the final model.

$$\underbrace{\begin{bmatrix} \mathbf{y}_1\\ \mathbf{y}_2\\ \mathbf{y}_3\\ \mathbf{y}_4\end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix}\mathbf{F}\\ \mathbf{F}\\ \mathbf{F}\\ \mathbf{F}\end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix}}_{\beta} + \underbrace{\begin{bmatrix} \mathbf{S}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{I}\\ \mathbf{0}&\mathbf{S}&\mathbf{0}&\mathbf{0}&\mathbf{I}\\ \mathbf{0}&\mathbf{0}&\mathbf{S}&\mathbf{0}&\mathbf{I}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{S}&\mathbf{I}\end{bmatrix}}_{\mathbf{Z}} \underbrace{\begin{bmatrix} b_1^{subject}\\ b_1^{slope}\\ \hdashline b_2^{subject}\\ b_2^{slope}\\ \hdashline b_3^{subject}\\ b_3^{slope}\\ \hdashline b_4^{subject}\\ b_4^{slope}\\ \hdashline b_1^{item}\\ b_2^{item}\\ b_3^{item}\\ b_4^{item}\end{bmatrix}}_{\mathbf{b}} + \underbrace{\begin{bmatrix}\epsilon_1\\ \epsilon_2\\ \epsilon_3\\ \epsilon_4\end{bmatrix}}_{\epsilon} .$$

Because every subject has to respond to every item, the final fixed effect design matrix

$\mathbf{X}$ is simply a stack of four $\mathbf{F}$ matrices. For the random effects, we have to ensure that each subject receives its intercept and slope parameter. Therefore, we have to extend the vector $\mathbf{b}$ and impose a block diagonal structure for the subject-dependent random effects in the random effects design matrix. To finalize the design matrix for the random effects $\mathbf{Z}$, we must extend each block row with a diagonal matrix $\mathbf{I}$.

We must specify the covariance matrix for the random effects ψ_θ and the error component $\mathbf{\Lambda}_\theta$ to complete the model. When we look at $\mathbf{b}$, we see that it is composed of four subject-specific blocks, each with a covariance matrix

$$\mathbf{\Sigma}_\theta{}^{subject} := \begin{bmatrix} \sigma^2_{subject} & \sigma_{subject,\, slope} \\ \sigma_{subject,\, slope} & \sigma^2_{slope} \end{bmatrix}),$$

And one block for the items. By design of the experiment, the items are generated (or drawn in probabilistic parlance) independently. For the multivariate normal distribution, the independence of components is equivalent to zero covariance between the components. Thus the covariance matrix for the item block of $\mathbf{b}$ looks like

$$\mathbf{\Sigma}_\theta{}^{item} := \begin{bmatrix} \sigma^2_{item} & 0 & 0 & 0 \\ 0 & \sigma^2_{item} & 0 & 0 \\ 0 & 0 & \sigma^2_{item} & 0 \\ 0 & 0 & 0 & \sigma^2_{item} \end{bmatrix}.$$

By design of the experiment, the subjects are also independent of each other. Thus, we can stack the individual covariance matrices together in a block diagonal fashion.

$$\psi_\theta = \begin{bmatrix} \mathbf{\Sigma}_\theta{}^{subject} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_\theta{}^{subject} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Sigma}_\theta{}^{subject} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{\Sigma}_\theta{}^{subject} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{\Sigma}_\theta{}^{item} \end{bmatrix}$$

as the covariance matrix for $\mathbf{b}$. The last model component we must specify is the covariance matrix for ϵ. Recall that we had to introduce random effects in the model to ensure independence between observations so that we can now make an i.i.d. assumption for $\epsilon_{\mathbf{si}}$. Consequently, the covariance matrix for the error component is

$$\mathbf{\Lambda}_\theta = \sigma^2_{error} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}.$$

A final look at all the covariance matrices shows us that they are determined by the four terms σ^2_{error}, $\sigma^2_{subject}$, σ^2_{slope}, σ^2_{item} and $\sigma_{subject,\, slope}$. Putting them together in a single parameter vector $\theta = (\sigma^2_{error}, \sigma^2_{subject}, \sigma^2_{slope}, \sigma^2_{item}, \sigma_{subject,\, slope})^\top$ finalizes our description.

2.2.2 Linear Mixed Effects Models: Fitting a Model to Data (Parameter Optimization)

In principle, there are two ways to calculate maximum likelihood estimators for an LMEM. First, we present a conceptually simple approach based on the distribution $p(\mathbf{Y}|\boldsymbol{\beta}, \theta)$. Let us assume that θ is known, so that $\mathbf{V} = \mathbf{Z}\psi_\theta\mathbf{Z}^\top + \mathbf{\Lambda}_\theta$ is known, then

$$p(\mathbf{Y}|\boldsymbol{\beta}, \theta) = \frac{1}{\sqrt{|\mathbf{V}|(2\pi)^N}} \exp(-\frac{1}{2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top \mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})).$$

The maximum likelihood estimator is found by optimizing the log-likelihood objective (terms and factors not involving β are dropped)

$$\ell(\beta) = -\frac{1}{2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top \mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}).$$

This is a simple convex optimization problem with the solution.

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}^\top \mathbf{V}^{-1}\mathbf{Y}.$$

If we want to obtain estimates (also called predictions) for the random effects, we estimate $\mathbb{E}_{\mathbf{b}|\mathbf{Y}=\mathbf{y}}[\mathbf{b}]$. Recall that the joint distribution of $\mathbf{b}$ and $\mathbf{Y}$ is

$$\begin{bmatrix}\mathbf{b}\\\mathbf{Y}\end{bmatrix} \sim \mathcal{N}(\begin{bmatrix}\mathbf{0}\\\mathbf{X}\beta\end{bmatrix}, \begin{bmatrix}\psi_\theta & \psi_\theta\mathbf{Z}^\top\\\mathbf{Z}\psi_\theta & \mathbf{V}\end{bmatrix}),$$

Which yields for the conditional expectation of $\mathbf{b}$ given $\mathbf{Y} = \mathbf{y}$, by definition of the conditional expectation of multivariate Gaussians, the following expression:

$$\mathbb{E}_{\mathbf{b}|\mathbf{Y}=\mathbf{y}}[\mathbf{b}] = (\psi_\theta\mathbf{Z}^\top\mathbf{V}^{-1})(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}).$$

Substituting $\hat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$ we obtain the following estimator for $\mathbf{b}$:

$$\hat{\mathbf{b}} = (\psi_\theta\mathbf{Z}^\top\mathbf{V}^{-1})(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}).$$

The estimated values obtained via the approach above are identical to the estimates of a more complex estimator called the Henderson equations (or mixed model equations, see [Hen+59]). They are based on the distribution $p(\mathbf{Y}|\mathbf{b}, \boldsymbol{\beta}, \theta)$ and allow estimating β and $\mathbf{b}$ simultaneously, and assume in general that θ is unknown. The advantage of the Henderson equations is that they allow a computationally more efficient estimation since only the inversion of matrices of much smaller dimensions than $\mathbf{V}$ is required. In general, when θ is unknown, it needs to be replaced by an estimator. There are a variety of different techniques to do so, and the inclined reader is referred to [PB00; MS01; WWG07; Dem13; Woo17] for an extensive elaboration of these.

2.2.3 Linear Mixed Effects Models: Statistical Inference (Likelihood Ratio Statistic)

Score Function and Fisher Information

A key concept for likelihood-based statistics is the *score function.* Let Y be a random variable distributed according to $p_\theta(y)$, and let $\ell(\theta) := \log p_\theta(y)$. Then the score function is defined as

$$S(\theta) := \frac{\partial}{\partial\theta}\ell(\theta).$$

The maximum likelihood estimator $\hat{\theta}$ is thus the solution to the *score equation* which is defined as

$$S(\theta = \hat{\theta}) = 0. \tag{2.3}$$

The *Fisher information* $I(\theta)$ of Y is defined as

$$I(\theta) := \mathbb{E}_\theta[S(\theta)^2] = \int S(\theta)^2 p_\theta(y)dy. \tag{2.4}$$

Under the mild assumption that the order of integration and differentiation can be exchanged, $I(\theta)$ can be written as the variance of the score function:

$$I(\theta) := \mathbb{V}_\theta[S(\theta)]. \tag{2.5}$$

The derivation is given by the following calculations:

$$\begin{aligned}
\mathbb{E}_\theta\left[S(\theta)\right] &= \mathbb{E}\left[\frac{\partial}{\partial\theta}\ell(\theta)\right] \\
&= \int \left[\frac{\partial}{\partial\theta}\ell(\theta)\right] p_\theta(y)dy \\
&= \int \left[\frac{\partial}{\partial\theta}\log p_\theta(y)\right] p_\theta(y)dy \\
&= \int \left[\frac{\frac{\partial}{\partial\theta}p_\theta(y)}{p_\theta(y)}\right] p_\theta(y)dy \\
&= \frac{\partial}{\partial\theta}\int p_\theta(y)dy \\
&= \frac{\partial}{\partial\theta}1 = 0.
\end{aligned}$$

The equivalence of equations (2.5) and (2.4) follows since

$$\begin{aligned}\mathbb{V}_\theta[S(\theta)] &= \mathbb{E}_\theta[S(\theta)^2] - \mathbb{E}_\theta[S(\theta)]^2\\ &= \mathbb{E}_\theta[S(\theta)^2] - 0.\end{aligned}$$

Given that $\ell(\theta) := \log p_\theta(y)$ is twice differentiable in θ, another useful equivalence of $I(\theta)$ can be shown:

$$I(\theta) = -\mathbb{E}_\theta[\frac{\partial^2}{\partial\theta^2}\ell(\theta)]. \tag{2.6}$$

The second derivative of $\ell(\theta)$ is

$$\begin{aligned}\frac{\partial^2}{\partial\theta^2}\ell(\theta) &= \frac{\partial^2}{\partial\theta^2}\log p_\theta(y)\\ &= \frac{p_\theta(y)\frac{\partial^2}{\partial\theta^2}p_\theta(y) - \left[\frac{\partial}{\partial\theta}p_\theta(y)\right]^2}{\left[p_\theta(y)\right]^2}\\ &= \frac{\frac{\partial^2}{\partial\theta^2}p_\theta(y)}{p_\theta(y)} - \left[\frac{\partial}{\partial\theta}\ell(\theta)\right]^2\\ &= \frac{\frac{\partial^2}{\partial\theta^2}p_\theta(y)}{p_\theta(y)} - \left[S(\theta)\right]^2.\end{aligned}$$

Taking expectations and assuming that the order of integration and differentiation can be exchanged, we see that the first term cancels, and we end up with an equivalence of equations (2.6) and (2.4):

$$\begin{aligned}\mathbb{E}_\theta\left[\frac{\partial^2}{\partial\theta^2}\ell(\theta)\right] &= \int\left[\frac{\partial^2}{\partial\theta^2}p_\theta(y)\right]dy - I(\theta)\\ &= 0 - I(\theta).\end{aligned}$$

Taylor Expansion and Asymptotic Distribution

Assume $\theta_0, \hat{\theta} \in \mathbb{R}$ to be scalars indicating our null and alternative hypotheses, respectively. Furthermore, assume a random variable Y with distribution $p_\theta(y)$. The likelihood ratio statistic can then be written as follows:

$$W = -2\log\Lambda = -2\log\frac{p_{\theta_0}(y)}{p_{\hat{\theta}}(y)} = 2\left(\ell(\hat{\theta}) - \ell(\theta_0)\right).$$

The central argument employed in [Wil38] is to replace $\ell(\theta_0)$ by its quadratic Taylor expansion around the maximum likelihood estimator $\hat{\theta}$. Let us first consider the case of a single observed sample point:

$$\begin{aligned}
W &= 2\left(\ell(\hat{\theta}) - \ell(\theta_0)\right) \\
&\approx 2\left(\ell(\hat{\theta}) - \ell(\hat{\theta}) - (\theta_0 - \hat{\theta})\frac{\partial}{\partial\theta}\ell(\hat{\theta}) - \frac{1}{2}(\theta_0 - \hat{\theta})^2\frac{\partial^2}{\partial\theta^2}\ell(\hat{\theta})\right) \\
&= (\hat{\theta} - \theta_0)^2\frac{\partial^2}{\partial\theta^2}\ell(\hat{\theta}).
\end{aligned}$$

The result follows the score equation (2.3). For N i.i.d observations, $\ell(\hat{\theta}_N) = \sum_{n=1}^{N}\ell_{y_i}(\hat{\theta}_N) := \sum_{n=1}^{N}\log p_{\hat{\theta}_N}(y_i)$. Hence, we get the following approximation:

$$\begin{aligned}
W &\approx (\hat{\theta}_N - \theta_0)^2\sum_{n=1}^{N}\frac{\partial^2}{\partial\theta^2}\ell_{y_i}(\hat{\theta}_N) \\
&= \left(\sqrt{N}(\hat{\theta}_N - \theta_0)\right)^2\left(\frac{1}{N}\sum_{n=1}^{N}\frac{\partial^2}{\partial\theta^2}\ell_{y_i}(\hat{\theta}_N)\right) \\
&\xrightarrow[n\to\infty]{p} \left(\sqrt{N}(\hat{\theta}_N - \theta_0)\right)^2 I(\theta_0).
\end{aligned}$$

The result follows since the empirical Fisher information converges in probability to the Fisher Information Matrix $I(\theta_0)$. An application of Theorem 2.2 then lets us state the asymptotic distribution of the likelihood ratio statistic as follows:

$$W \overset{app}{\sim} \chi^2_{df=1}.$$

For more information on likelihood-based statistical methods and related asymptotic results, the reader is referred to [Paw01; Vaa98; Dav03].

The Generalized Likelihood Ratio Test

The hypotheses to be tested in a likelihood ratio test are hypotheses about the parameters of probability distributions. Suppose we observe a sample $Y = (Y_1, Y_2, \ldots, Y_N)$ from a probability distribution p_θ, and we wish to test the null hypothesis

$$\mathrm{H}_0 : \theta \in \Theta_0,$$

against the alternative hypothesis

$$\mathrm{H}_1 : \theta \in \Theta_1.$$

If both hypotheses consist of single points θ_0 and θ_1, then a most powerful test can be based on the test statistic of the likelihood ratio.

$$\frac{\prod_{i=1}^{N} p_{\theta_0}(Y_i)}{\prod_{i=1}^{N} p_{\theta_1}(Y_i)},$$

by the Neyman-Pearson lemma [NP33].

An extension of the Neyman-Pearson theory replaces single points by the supremum over a restricted parameter space Θ_0 for the null hypothesis and by the supremum over the whole parameter space $\Theta = \Theta_0 \cup \Theta_1$ for the alternative hypothesis, leading to the generalized likelihood ratio statistic.

$$\frac{\sup_{\theta \in \Theta_0} \prod_{i=1}^{N} p_\theta(Y_i)}{\sup_{\theta \in \Theta} \prod_{i=1}^{N} p_\theta(Y_i)} = \frac{l_0}{l_1},$$

That builds the basis of the generalized likelihood ratio test. [LM12] describe the test in the following brief form:

Definition 2.1 (Generalized Likelihood Ratio Test (GLRT))**.**
Reject H_0 if the generalized likelihood ratio statistic

$$\lambda = \frac{l_0}{l_1},$$

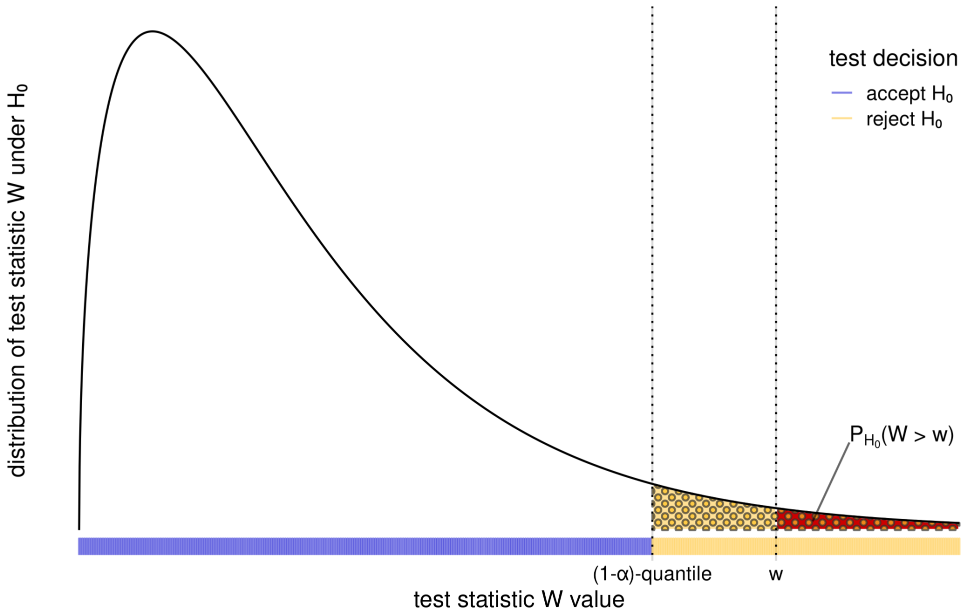

Figure 2.3: p-value based on χ^2 distribution.

has a value

$$0 < \lambda \leq \lambda^*$$

where λ^* is chosen such that $P(0 < \lambda \leq \lambda^* | \mathrm{H}_0 \text{ is true }) = \alpha$ for a significance level α.

The null hypothesis of the GLRT is the assumption that the restricted model l_0 explains the data adequately. Since $0 < \lambda \leq 1$, the intuition behind the test is that values of λ close to 1 suggest that the restricted model assumed under H_0 explains the data as well as a more complex model assumed under H_1. Thus H_0 should be accepted for such values of λ. Conversely, values of λ close to 0 suggest that the data are incompatible with the parameter values in the restricted model. Thus H_0 should be rejected in favor of H_1, which more adequately explains the data.

To determine the critical value λ^* for a given significance level α, we need to know the distribution of the test statistic λ. Fortunately, our test statistic is based on maximum likelihood estimates of parameters of a probability distribution — in our case, we will employ the parametric family of LMEMs that we already used for reliability assessment in Chapter 3 — and we can fall back on an asymptotic result similar to the Central Limit Theorem, this time a theorem showing the asymptotic normality of maximum likelihood estimates.[23]

Theorem 2.2 (Asymptotic Distribution of Maximum Likelihood Estimators)**.** Let $Y = (Y_1, Y_2, \dots, Y_N)$ be sample from a probability distribution p_θ, and define the log-likelihood of the sample as $\ell_N(\theta) = \log \prod_{i=1}^{N} p_\theta(Y_i)$. If the maximum likelihood estimator $\hat{\theta}$ exists as the solution to the equation $\frac{\partial}{\partial\theta}\ell_N(\theta) = 0$, in addition to second and third derivatives of $\ell_N(\theta)$, then the asymptotic distribution of $[N \cdot I_N(\theta)]^{1/2}(\hat{\theta} - \theta)$ is the standard normal distribution, where $I_N(\theta) = \mathbb{E}_{p_\theta}[(\frac{\partial}{\partial\theta} l_N(\theta))^2]$ is the Fisher information of the sample Y about θ.

Similar to the Central Limit Theorem 2.1, we consequently get a statement on the approximate distribution of $\hat{\theta}$ being the multivariate normal distribution with mean θ and variance $[N \cdot I_N(\theta)]^{-1}$:

$$\hat{\theta} \overset{app}{\sim} \mathcal{N}(\theta, [N \cdot I_N(\theta)]^{-1}). \tag{2.7}$$

Using Theorem 2.2, it can be shown that under the null hypothesis, $-2\log\lambda$ follows a χ^2 distribution. This result is due to [Wil38]. We present a derivation of the asymptotic distribution of the likelihood ratio statistic for the simple case of a single random variable Y and a scalar-valued parameter θ in Appendix 2.2.3.[24] In short, the result states that the random variable W, defined as

$$W = -2\log\Lambda = 2\log\frac{l_1(Y_1, \dots, Y_N)}{l_0(Y_1, \dots, Y_N)} \overset{app}{\sim} \chi^2_{df=k_1-k_0}, \tag{2.8}$$

follows a χ^2 distribution with $k_1 - k_0$ degrees of freedom if the general model yielding l_1 has k_1 parameters and the restricted model yielding l_0 has k_0 parameters. This allows us to reject H_0 if the observed value w of W is greater than the $(1-\alpha)$-quantile of the aforementioned distribution, that is if the p-value

$$p \coloneqq P_{\mathrm{H}_0}(W > w) \tag{2.9}$$

[23] Derivations and proofs for variants of the asymptotic argument can be found in [Vaa98], Chapter 7.

[24] A detailed proof is given in [Vaa98], Chapter 16.

It is smaller than the rejection level α. The critical region of the χ^2 distribution is illustrated in Figure 2.3. Again, since the p-values can be calculated directly, it is good practice to report the p-value instead of assessing the statistical significance at a given α-level [McS+19].

2.3 Model Based Algorithm Comparison: Analyzing an Example

In this section, I will demonstrate and explain LMEM-based algorithm comparison by reanalyzing an experiment on interactive machine translation conducted by [KBR20]. This section aims to show how the above-discussed tools and theorems can be combined to create a powerful (in the statistical sense) method to analyze the complex evaluation data generated during a machine learning experiment.

In the study above, the researchers wanted to improve the performance of a pre-trained machine translation system by incorporating human feedback in a reinforcement learning mechanism. They investigated two modes of correction. The first mode is called "Marking". In this mode, the annotators mark the wrong words. The feedback is incorporated into the objective function to maximize the objective when the probability of the translated sequence's correct (non-marked) tokens is increased and decreased for the incorrect (marked) tokens. The second mode is called "Post Edit". In this mode, the annotators corrected the translations. The updated translation is then used as a new gold standard translation, and the system is trained using these new gold standards. The quality of machine-translated sentences was obtained by calculating TER, BLEU, and METEOR scores relative to the original gold standard translation. To avoid redundancy, we limit our showcase to TER evaluation. The fine-tuning process was replicated using three different initial random seeds –but keeping all other meta-parameters equal as it is standard for fine-tuning– for Marking and PostEdit feedback signals.

The apparent research question is, "Which feedback method improves over the baseline? If so, which shows the largest gain?". To answer these questions, Kreutzer et al. collected evaluation data by applying the baseline and the models fine-tuned[25] on marking and post-edit feedback on 1,041 test sentences.[26].

[25] In their original paper, Kreuter et al. considered only the best models based on a descriptive ranking based on the development set estimates of the out-of-sample risk for each fine-tuning method for evaluation. But for the analysis presented in this section, we consider the evaluation data for all seven models.

[26] Note that two of the 1,043 test sentences reported in [KBR20] were duplicates that we removed in

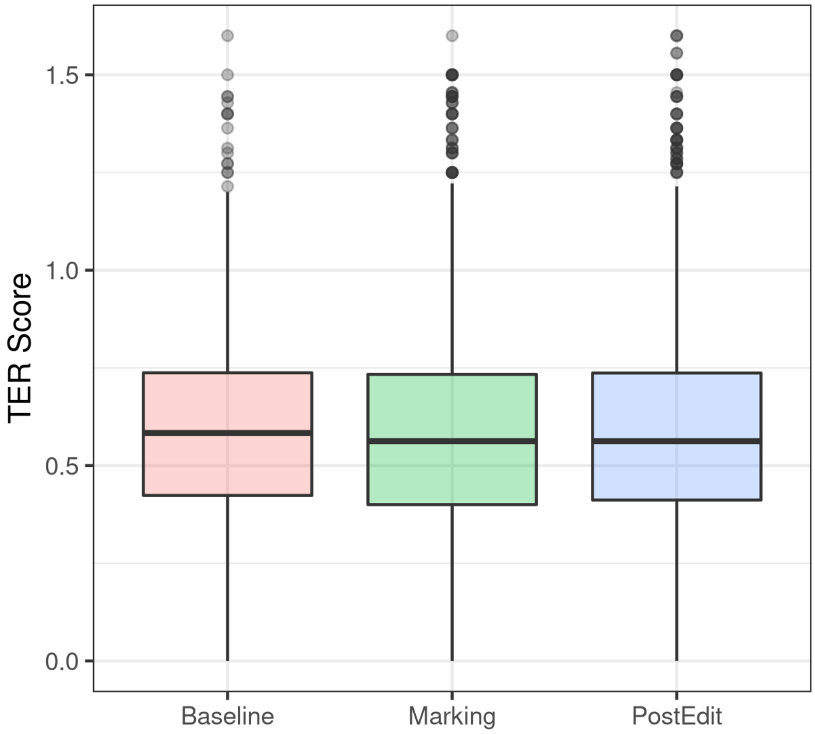

Figure 2.4: Median TER scores for baseline and machine translation systems fine-tuned on markings or post-edits.

Let us first look at the TER evaluation score in a boxplot shown in Figure 2.4.

The horizontal line in the middle of the box marks the median value of the data points in the specific group. The box indicates the range where the middle 50% of the data points are located. The vertical lines are called whiskers and identify observations with unusually large or small values in the data set (so-called outliers), represented by point-like symbols below or above the whisker. Figure 2.4 shows that the shape of the box plot is somewhat similar for all three systems, with the boxplots for "Marking" and "PostEdit" being located slightly below the "Baseline" boxplot, meaning that by central tendency, both feedback methods yield slightly improved translation quality.

Let us first analyze the statistical significance of the observed evaluation results. Since each of the fine-tuning models was trained three times with different random seeds, in a first approach following [Hot+05], we average the TER scores for the models trained on human annotations and assess the statistical significance of the average result differences to the baseline results. The averaged data then can be modeled by an LMEM that equals

our LMEM experiments.

a standard linear model as in (2.10) fitted to the sentence-wise averaged[27] evaluation data. I will use this discussion to illustrate the following aspects of model-based analysis:

- Explain the fixed effect structure and show how the categorical variable algorithm is encoded in the analysis model.
- Show the mathematical meaning and the interpretation of the model parameters associated with the fixed effects.
- Illustrate the nested models setup and the application of GLRT for an omnibus algorithm comparison.

Before we specify the model that allows us to analyze the evaluation data, let us first fix some basic ideas about it. Firstly, we assume that the valuation data is a sample obtained by querying an abstract probabilistic process. This process is a mixture of three potentially different TER score distributions, one for each algorithm. The following model expresses this simple idea:

$$\text{TER} = \underbrace{\beta_0 + \beta_{\text{marking}} \cdot \mathbb{I}_{\{\text{marking}\}}\left(\text{System}\right) + \beta_{\text{postEdit}} \cdot \mathbb{I}_{\{\text{postEdit}\}}\left(\text{System}\right)}_{\text{fixed effects}} + \underbrace{\epsilon}_{\text{error}} \tag{2.10}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and $\mathbb{I}_{\{\text{condition}\}}\left(\text{variable}\right)$ is an indicator function that returns one if variable meets the condition and zero else. For a better understanding and a proper interpretation of the model and its parameters, let us consider the expected value of TER based on this model:

$$\mathbb{E}\left[\text{TER}|\text{Algorithm}\right] = \beta_0 + \beta_{\text{marking}} \cdot \mathbb{I}_{\{\text{marking}\}}\left(\text{System}\right) + \beta_{\text{postEdit}} \cdot \mathbb{I}_{\{\text{postEdit}\}}\left(\text{System}\right)$$

We see that the expected value of TER is equivalent to the fixed effect part of the model, which is a function depending on the value of Algorithm and the weights β_0, $\beta_{marking}$ and $\beta_{postEdit}$. Thus conditional on the value of Algorithm, the model returns different values for the expected value of TER. Let $\mathbb{E}\left[\text{TER}|\text{Algorithm} = a\right]$ denotes the expected value of TER for algorithm a. Let Algorithm = baseline than

$$\mathbb{E}\left[\text{TER}|\text{baseline}\right] = \beta_0.$$

[27] The average is calculated for each sentence within each system.

Thus β_0 (the model intercept) represents the baseline algorithm's expected TER score. Hence the estimator $\hat{\beta}_0$ is an estimator of the expected out-of-sample TER score of the baseline algorithm. Now that we know how to interpret the intercept, let us move forward and consider the case Algorithm = Marking. The expected TER value for marking is:

$$\mathbb{E}\left[\text{TER}|\text{marking}\right] = \beta_0 + \beta_{marking}.$$

Rearranging this equality and using the previous result for β_0, we get:

$$\beta_{marking} = \mathbb{E}\left[\text{TER}|\text{marking}\right] - \beta_0 = \mathbb{E}\left[\text{TER}|\text{marking}\right] - \mathbb{E}\left[\text{TER}|\text{baseline}\right].$$

Hence $\beta_{marking}$ represents the difference between the expected values of the marking and the baseline algorithm. By extension, the estimator $\hat{\beta}_{marking}$e is an estimator for the differences between the expected out-of-sample risks of both systems. An analogous result for $\beta_{postEdit}$ can be obtained. In summary, we have seen that the fixed effect coefficients of this particular linear mixed effects model can be interpreted as either expected risks or differences of expected risks. Thus the maximum likelihood estimators of this model's parameters provide consistent and asymptotic efficient estimators of the expected risk of an algorithm (or classifier). Further, maximum likelihood theory also provides asymptotic distributional results facilitating statistical inference for these estimators. The first hypothesis pair we want to test is

$$\begin{aligned}
\text{H}_0\text{:}\quad & \mathbb{E}\left[\text{TER}|\text{baseline}\right] = \mathbb{E}\left[\text{TER}|\text{marking}\right] = \mathbb{E}\left[\text{TER}|\text{postEdit}\right]\\
\text{H}_1\text{:}\quad & \mathbb{E}\left[\text{TER}|\text{baseline}\right] \neq \mathbb{E}\left[\text{TER}|\text{marking}\right]\\
& \vee\ \mathbb{E}\left[\text{TER}|\text{baseline}\right] \neq \mathbb{E}\left[\text{TER}|\text{postEdit}\right]\\
& \vee\ \mathbb{E}\left[\text{TER}|\text{marking}\right] \neq \mathbb{E}\left[\text{TER}|\text{postEdit}\right]
\end{aligned}$$

This test is called an *omnibus test.* It tests the *null hypothesis* H_0 of equal expected risks for all algorithms against the *alternative hypothesis* H_1 that at least one pair of algorithms has unequal expected risks. The corresponding hypothesis pair, expressed in terms of fixed effect parameters, is:

$$\begin{aligned}
\mathrm{H}_0\colon \quad & \beta_{marking} = \beta_{postEdit} = 0 \\
\mathrm{H}_1\colon \quad & \beta_{marking} \neq 0 \\
& \quad \vee \ \beta_{postEdit} \neq 0 \\
& \quad \vee \ \beta_{marking} \neq \beta_{postEdit}
\end{aligned}$$

The model specified by (2.10) corresponds to hypothesis H_0, because it imposes no restriction on the value of β_0, $\beta_{marking}$ or $\beta_{postEdit}$. To apply the GLRT to decide whether we should accept H_0 or H_1, we need a model that corresponds to H_0. We have seen, that the null hypothesis states that $\beta_{marking} = 0$ and $\beta_{postEdit} = 0$, thus the corresponding model is

$$\mathrm{TER} = \underbrace{\beta_0}_{\text{fixed effects}} + \underbrace{\epsilon}_{\text{error}}. \tag{2.11}$$

Under this model, the expected value of TER for all systems is

$$\mathbb{E}\left[\mathrm{TER}\right] = \beta_0.$$

The model (2.11) is a particular case of the model (2.10) because we arrived at it by restricting the values of $\beta_{marking}$ and $\beta_{postEdit}$ to 0. Such a relation between two models is called *nested*, and one says that model (2.11) is nested in the model (2.10). This setup of *nested models* [PB00] allows us to conduct a GLRT with the restricted model (2.11) representing the null hypothesis, the more general model (2.10) representing the alternative hypothesis.

Applying this technique to the data from [KBR20], a GLRT which compares model (2.13) against the restricted model (2.11) yields a p-value of 0.517. According to a standard significance level of 0.05, this result is too high to reject the null hypothesis that the three algorithms have equal performance. Do we have to conclude that the difference in performance evaluations between the three algorithms is not statistically significant? Or was this simple and often recommended strategy to average over algorithm instances not powerful (in the statistical sense) enough?

Instead of trimming the data to fit a standard method, let us adapt the method to the data. The major obstacle to keeping all observations for each algorithm in the evaluation data is introduced by the reasonable assumption that if the expected risk of the algorithms involved in the experiment is different, then the evaluation scores of a particular test sentence obtained by the instances of one algorithm will be more similar than evaluation scores obtained by instances of different algorithms. Hence, the data could not be considered to be iid distributed. We can adapt model (2.10) to this clustered data structure by realizing that if we expand the model so that it conditions not only on the algorithm but also on the input sentence, then errors would be independent and identically distributed again if we assume that the meta-parameter configurations are determined before the experiment[28]. Of course, such a model would be very parameter intensive and, more importantly, model a quantity the experimenter is not interested in. LMEMs can circumvent this problem by using random intercepts. Random intercepts are a form of random effect allowing the modeler to "condition" the distribution on the input sentence in a way that doesn't affect the fixed effect structure of the model. The LMEM corresponding to this adaption of model (2.10) is:

$$\begin{aligned} \text{TER} = &\underbrace{\beta_0 + \beta_{\text{marking}} \cdot \mathbb{I}_{\{\text{marking}\}}\left(\text{System}\right) + \beta_{\text{postEdit}} \cdot \mathbb{I}_{\{\text{postEdit}\}}\left(\text{System}\right)}_{\text{fixed effects}} \\ &+ \underbrace{b^{input\_id}}_{\text{random intercept}} + \underbrace{\epsilon}_{\text{error}} \end{aligned} \tag{2.12}$$

where $b^{input_id} \sim \mathcal{N}(0, \sigma^2_{input\_id})$. Because we haven't changed the fixed effect structure, this models expectation is still:

$$\mathbb{E}\left[\text{TER}|\text{Algorithm}\right] = \beta_0 + \beta_{\text{marking}} \cdot \mathbb{I}_{\{\text{marking}\}}\left(\text{System}\right) + \beta_{\text{postEdit}} \cdot \mathbb{I}_{\{\text{postEdit}\}}\left(\text{System}\right).$$

But in contrast to model (2.10) where $\mathbb{V}\left[\text{TER}\right] = \sigma^2$ the variance of TER is now decomposed into

$$\mathbb{V}\left[\text{TER}\right] = \sigma^2_{input_id} + \sigma^2.$$

Based on this equation, we can draw the following conclusions:

- If the clustering within the evaluation scores is negligible (thus $\sigma^2_{input_id} \approx 0$, the specification of (2.10) is essentially the same as (2.12) but the latter consumes one

[28] This assumption is equivalent to assuming that the meta-parameter configurations are drawn independently from the meta-parameter space.

Table 2.2: Effect of evaluation strategy on estimated error variance and significance.

Strategy	**H$_1$ model**	**H$_0$ model**	$\hat{\sigma}^2$	p**-value**
average replications per random seed	Eq. (2.10)	Eq. (2.11)	0.2576	0.517
group replications at sentence level	Eq. (2.12)	Eq. (2.13)	0.0591	< 0.0001

more parameter.

- The more pronounced the clustering of the evaluation scores are, the larger $\sigma^2_{input_id}$ will be compared to σ^2, and consequently, the more powerful a GLRT based on this model will be.

If the model's parameters (2.12) are estimated via the maximum likelihood principle, one can use a GLRT to test the hypothesis about the fixed effect parameters. The proper nested model for an omnibus test of (2.12) is

$$\text{TER} = \underbrace{\beta_0}_{\text{fixed effects}} + b^{input\_id} + \underbrace{\epsilon}_{\text{error}}. \tag{2.13}$$

Changing the model so that it fits the structure of the evaluation data by adding the random effect ν_s reduces the estimated residual error σ^2 from 0.2576 to 0.0591 indicating a high clustering in the data, resulting in a p-value of < 0.0001 for a comparison of models (2.12) to (2.13). Thus we can reject the null hypothesis and assume that at least two algorithms have different expected risks.

The results of this section are summarized in 2.2. The overall comparison would be finished now if the experiment had only involved two competing algorithms. Because the disjunctive formula of the alternative hypothesis would be reduced to an atomic formula about the inequality of two expected risks and therefore it will be clear which pair of algorithms have none-equal expected risks. But once the experiment comprises more than two competitors, like in the current example, a significant omnibus test must be followed by a so-called posthoc analysis to arrive at a more detailed statement about the relations of the algorithm's expected risks.

Posthoc Analysis for Kreutzer et al.

Several ways exist to conduct a post hoc analysis of the parameters of an LMEM. I will present an approach that conducts all pairwise comparisons utilizing GLRTs to test appropriate sub-hypothesis. The first sub-hypothesis pair that we want to test is:

$$\begin{aligned} \mathrm{H}_0\colon &\quad \mathbb{E}\left[\mathrm{TER}|\mathrm{baseline}\right] = \mathbb{E}\left[\mathrm{TER}|\mathrm{marking}\right] \\ \mathrm{H}_1\colon &\quad \mathbb{E}\left[\mathrm{TER}|\mathrm{baseline}\right] \neq \mathbb{E}\left[\mathrm{TER}|\mathrm{marking}\right] \end{aligned}$$

This pair corresponds to a direct comparison of the baseline and the marking-enhanced algorithm. As previously discussed, we need an appropriate data model corresponding to the null hypothesis and one connected to the alternative hypothesis. In complete analogy to the models involved for the omnibus test, the alternative hypothesis model is

$$\mathrm{TER} = \underbrace{\beta_0 + \beta_{\mathrm{marking}} \cdot \mathbb{I}_{\{\mathrm{marking}\}}\left(\mathrm{System}\right)}_{\text{fixed effects}} + \underbrace{b^{input\_id}}_{\text{random intercept}} + \underbrace{\epsilon}_{\text{error}} \tag{2.14}$$

and for the null hypothesis

$$\mathrm{TER} = \underbrace{\beta_0}_{\text{fixed effects}} + \underbrace{b^{input\_id}}_{\text{random intercept}} + \underbrace{\epsilon}_{\text{error}} \tag{2.15}$$

Both models are fitted to a subset of the evaluation data, where the observations from the algorithms not included in the current comparison are removed, and a GLRT is conducted on the obtained likelihoods. This procedure is iterated analogously for all pairwise comparisons. The Bonferoni-Holm procedure is applied to adjust the resulting p-values for multiplicity. The result is summarized in Table 2.3. This posthoc analysis yields significant differences between baseline and fine-tuning on markings ($p < 0.0001$), between baseline and fine-tuning on post-edits ($p < 0.0001$), but no significant difference between fine-tuning on markings and fine-tuning on post-edits ($p = 0.0685$). Thus we conclude that both feedback methods improve the baseline model, but neither is better.

Table 2.3: p-values for pairwise TER differences between systems on the test set.

	p-value
baseline - marking	< 0.0001
baseline - post-edit	< 0.0001
marking - post-edit	0.0685

Algorithm Comparsion conditional on Data Properties

In a further step, we will investigate if the result of the previous section is homogeneous for all inputs. In machine translation, it is often the case that models work better for short inputs than for longer ones. Thus, it would be interesting to investigate if the involved algorithms in this experiment perform homo- or heterogeneously concerning this input property. To get a first impression, we create a scatter plot with source sentence length on the abscissa and TER of the translation on the ordinate for all systems shown in Figure 2.5. These plots indicate that the contour lines of the point cloud are similar for all three algorithms, and the relation between TER and source sentence length is non-linear but monotonically increasing. We see an increase in TER for short sentences ($<$ 15 words), followed by a rather flat section for sentences of length $15-55$ words, and a steep increase for very long sentences ($>$ 55 words). To emphasize this point, we classify the sentence length into three categories "short" (< 15), "typical" ($15 - 55$), and "very long" (> 55), and create boxplots of the data presented in Figure 2.6. This visual comparison highlights that while the three systems behave nearly identically for typical sentences, they show noticeable differences for short and very long sentences. Furthermore, it suggests that most of the improvement gained from human feedback happens for very long sentences, and to a lesser degree, for short ones.

To test this hypothesis, we extend the model (2.12) by including a fixed effect for sentence length and a fixed effect to analyze interactions between algorithm and sentence lengths, yielding the following model:

$$\begin{aligned}\text{TER} = &\underbrace{\beta_0 + \text{Algorithm} + \text{SentenceLength} + \text{Algorithm x SentenceLength}}_{\text{fixed effects}} \\ &+ \underbrace{b^{input\_id}}_{\text{random intercept}} + \underbrace{\epsilon}_{\text{error}}\end{aligned} \tag{2.16}$$

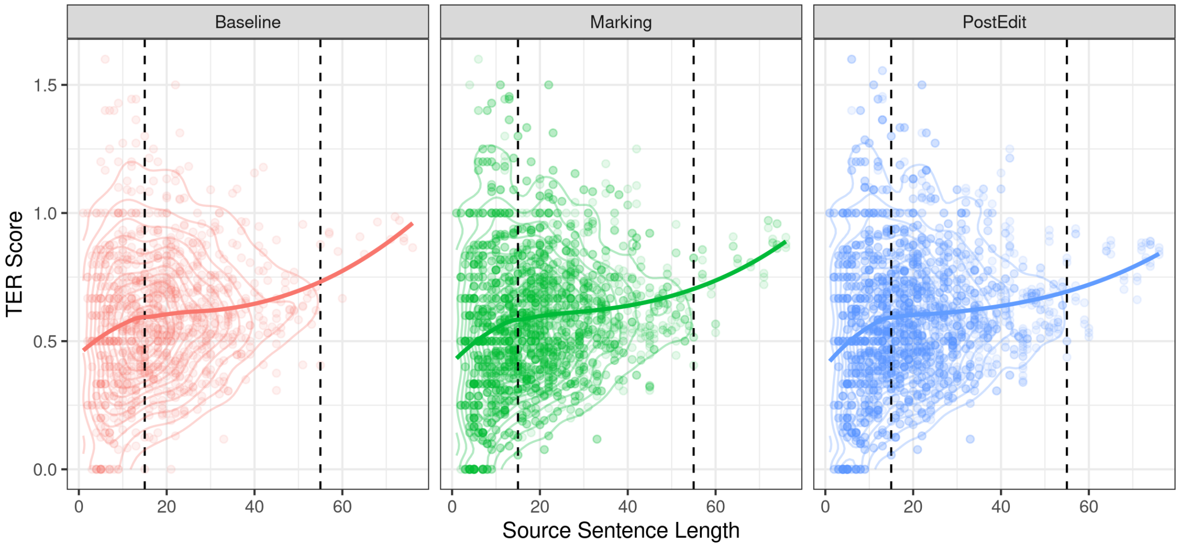

Figure 2.5: TER scores for baseline and machine translation systems fine-tuned on markings or post-edits, plotted against source sentence length.

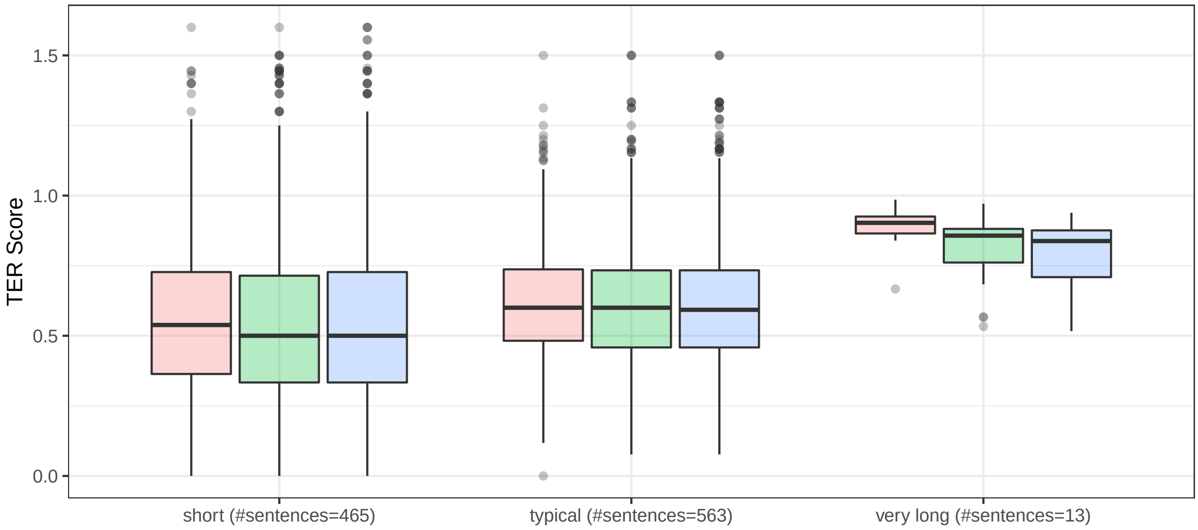

Figure 2.6: TER scores for baseline and machine translation systems fine-tuned on markings or post-edits, plotted against three bins of source sentence length.

where

$$
\begin{aligned}
\text{Algorithm} &:= \beta_{\text{marking}} \cdot \mathbb{I}_{\{\text{marking}\}}(\text{Algorithm}) \\
&\quad + \beta_{\text{postEdit}} \cdot \mathbb{I}_{\{\text{postEdit}\}}(\text{Algorithm}) \\
\text{SentenceLength} &:= \beta_{\text{typical}} \cdot \mathbb{I}_{\{\text{typical}\}}(\text{SentenceLength}) \\
&\quad + \beta_{\text{long}} \cdot \mathbb{I}_{\{\text{long}\}}(\text{SentenceLength})
\end{aligned}
$$

are called *main effects* of algorithm or sentence length and

$$\begin{aligned}&\text{Algorithm x SentenceLength} \coloneqq \\ &\quad \beta_{\text{typical,marking}} \cdot \mathbb{I}_{\{\text{typical}\}}\left(\text{SentenceLength}\right) \cdot \mathbb{I}_{\{\text{marking}\}}\left(\text{Algorithm}\right) + \\ &\quad \beta_{\text{long,marking}} \cdot \mathbb{I}_{\{\text{long}\}}\left(\text{SentenceLength}\right) \cdot \mathbb{I}_{\{\text{marking}\}}\left(\text{Algorithm}\right) + \\ &\quad \beta_{\text{typical,postEdit}} \cdot \mathbb{I}_{\{\text{typical}\}}\left(\text{SentenceLength}\right) \cdot \mathbb{I}_{\{\text{postEdit}\}}\left(\text{Algorithm}\right) + \\ &\quad \beta_{\text{long,postEdit}} \cdot \mathbb{I}_{\{\text{long}\}}\left(\text{SentenceLength}\right) \cdot \mathbb{I}_{\{\text{postEdit}\}}\left(\text{Algorithm}\right)\end{aligned}$$

is called the *interaction effect* algorithm and sentence length and is used to capture non-additive relationships between the regressors and the response variable. To illustrate the modeling effect of interaction terms, let us consider the expected TER score for the marking algorithm on typical long sentences:

$$\begin{aligned}&\mathbb{E}\left[\text{TER}|\text{Algorithm} = \text{marking}, \text{SentenceLength} = \text{typical}\right] = \\ &\qquad \beta_0 + \underbrace{\beta_{\text{marking}} + \beta_{\text{typical}}}_{\text{main effects}} + \underbrace{\beta_{\text{typical,marking}}}_{\text{interaction}} .\end{aligned}$$

This equation shows that the interaction can modify the purely additive composition of the main effects. We are interested in this modification because it allows us to analyze if the algorithms in the experiment compare differently across different input lengths. We start this assessment by testing the hypothesis pair:

$$\begin{aligned}\text{H}_0\colon\quad & \beta_{\text{typical,marking}} = 0 \\ & \wedge\ \beta_{\text{typical,postEdit}} = 0 \\ & \wedge\ \beta_{\text{long,marking}} = 0 \\ & \wedge\ \beta_{\text{long,postEdit}} = 0 \\ \text{H}_1\colon\quad & \beta_{\text{typical,marking}} \neq 0 \\ & \vee\ \beta_{\text{typical,postEdit}} \neq 0 \\ & \vee\ \beta_{\text{long,marking}} \neq 0 \\ & \vee\ \beta_{\text{long,postEdit}} \neq 0\end{aligned}$$

Which is an omnibus test of whether or not the data supports the presence of an interaction. Again we can use an GLRT were (2.16) corresponds to the alternative

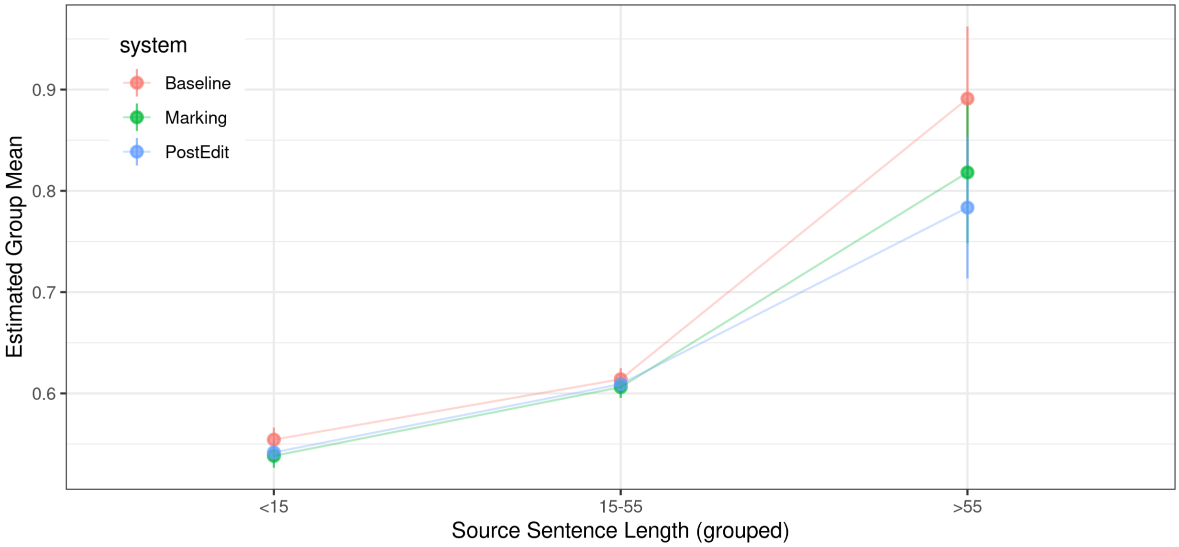

Figure 2.7: Interaction plot of estimated TER scores for baseline and machine translation systems fine-tuned on markings or post-edits and bins of source sentence length.

Table 2.4: p-values for pairwise differences between systems on source sentences of different lengths.

	short	**typical**	**very long**
baseline - marking	< 0.0001	0.0191	< 0.0001
baseline - post-edit	< 0.0001	0.1830	< 0.0001
marking - post-edit	0.1220	0.1830	0.0093

hypothesis and

$$\text{TER} = \underbrace{\beta_0 + \text{Algorithm} + \text{SentenceLength}}_{\text{fixed effects}} + \underbrace{b^{input\_id}}_{\text{random intercept}} + \underbrace{\epsilon}_{\text{error}} \tag{2.17}$$

To the null hypothesis. The GLRT conducted with both models shows a statistically significant p-value ($p < 0.0001$). Thus we reject the null hypothesis and investigate the interaction further. To this end, we visualize the estimated risks for each sub-group in a so-called interaction plot given in Figure 2.7 and perform pairwise comparisons similar to the posthoc analysis of the previous section between systems nested within source sentence length levels. The results of these comparisons are presented in Table 2.4.

In comparison to the result in Table 2.3 that we obtained without grouping sentences into length bins, Table 2.4 shows statistically significant differences between the estimated expected TER scores of marking and post-edit fine-tuning on long sentences. Furthermore, both systems significantly improve over the baseline model for short sentences. Still, for typical sentences, only the improvement of the marking system over the baseline is statistically significant. These results suggest that there is no uniform superiority of one feedback mode over the other and that an investigation into the interaction of feedback modes with data properties reveals essential patterns.

Discussion. The key practical feature of the proposed model-based approach is that it unifies special-purpose significance tests for particular evaluation metrics, meta-parameter variations, and multiple test data as proposed by [DSR19] or [Dro+17] into a single framework for hypothesis testing. In the presented example, we have demonstrated how LMEMs can simultaneously analyze all instances of algorithms obtained by a meta-parameter variation to compare algorithms and not specific algorithm instances. Because LMEMs are generalized linear models with the additional capacity of random effects, they can be used to analyze all response variables whose distribution is a member of the exponential family. This family includes many of the most common distributions, e.g., Gaussian, Bernoulli, categorical, exponential, gamma, Poisson, beta, etc. The evaluation scores of multiple test data sets can be concatenated, but one needs to make sure that the input identifiers are properly adjusted.

The idea of treating test data as random effects and thus increasing the power of statistical significance testing has already been proposed by [RK12] for information retrieval. However, the general applicability of LMEMs and GLRTs for significance testing under variations of meta-parameters and data properties has yet to be fully recognized in the wider NLP and data science research community.

A unique theoretical feature of the proposed model-based approach to significance testing is that it mutes the old question of which significance test is appropriate for which evaluation measure. In a model-based paradigm, one can handle the distributional properties of complex evaluation measures since they are not treated directly as test statistics of a significance test. Independent of the evaluation measure used, the test statistic of the GLRT is based on the parameter estimates of the LMEM trained on the evaluation data. It is a well-established theoretical result that the maximum likelihood parameter estimator asymptotically follows a normal distribution. Based on this fact, it can be shown that the generalized likelihood ratio test statistic asymptotically follows a

χ^2 distribution, which in turn allows to compute p-values for a wide range of hypotheses, including the typical A-B testing hypotheses pair. It should be mentioned that, while my presentation emphasized the GLRTs, other test statistics exist for hypothesis testing about the effect parameters of an LMEM, e.g., F-tests or wald -tests [PB00]. Especially the F-test is often the default implementation in most software packages because of its technical and computational convenience. But it must be mentioned that for analyzing experimental data from balanced designs (like the typical machine learning experiment), the GLRT and F-test (as well as the Wald-test) behave nearly identically.

Chapter 3

Algorithm Performance Analysis

One shortcoming of the contemporary state-of-the-art evaluation of machine learning experiments is its tendency to treat and analyze algorithms and classifiers in a black-box fashion. This attitude prohibits any empirically grounded insights into the relationship between an algorithm's performance concerning a task and the choice of its meta-parameters. This knowledge (especially when accumulated over several experiments) is of efficient value because it reduces the effort of the meta-parameter search a machine learning practitioner has to conduct to find a good classifier for her use case. Instead of running an extensive search over as large as possible space of meta-parameter combinations, she can focus her search on a smaller subset of the most promising meta-parameter variations. Thereby reducing the computational burden and, thus, the financial and ecological costs of implementation. Theoretically, such an evaluation can shed light on which meta-parameters are significant concerning an algorithm's performance and which are negligible and can be fixed.

An NLP subfield that has investigated a related question is information retrieval (IR), where systems are often compositions of several specific components. In this context, the question is determining how much each component contributes to the overall system performance and how the components interact. To this end, the performance of the IR system is approximately decomposed [Fer+15; Jay+15; RK12; TB95] as following:

$$\text{performance} \approx \underbrace{\text{topic effect}}_{\equiv\text{data effect}} + \underbrace{\text{system effect}}_{\equiv\text{meta-parameter effect}} + \underbrace{\text{topic} \times \text{system-interaction}}_{\equiv\text{data}\times\text{meta-parameter interaction}}$$

Based on this principle idea [FS16] suggested a full factorial experimental design (along with a GLMM to analyze the resulting evaluation data) to investigate the system effect. The idea of decomposing the variation of an observed value to quantify and study the contribution of potentially influential sources, being novel in the assessment of IR systems, is familiar. It was introduced by Fisher [Fis19] in the context of statistical genetics and further developed in the context of psychometrics [Bre01] to study the reliability of psychological and educational measurement instruments.

In this chapter, I want to investigate how techniques and concepts from these fields can be transferred and modified for more detailed algorithm analysis. To this end, I will treat machine learning models similar to human annotation or any other scientific instrument that assigns a measurement value to an object. In the case of machine learning, these measurements are either the outputs (predictions) of a classifier if its domain is numeric or an evaluation metric that measure the concordance of an output to a reference if the domain of the classifier is non-numeric like in chapter 2, my goal is to leverage the information already created during a machine learning experiment and not impose additional demands on the researcher. Nevertheless, it has to be mentioned that especially full-crossed factorial designs offer some technical benefits. In summary, in this chapter, I will

- Propose a novel approach to quantify the reliability of machine learning algorithms and data annotations.

- Show how LMEMs can be used to analyze the variance contribution of meta-parameters (and their interaction) for a wide range of rather arbitrary experimental designs.

- Show how LMEMs can be used to study meta-parameter/data property interactions.

But before I start my exposition, I want to discuss some of the terminology related to these topics. Then I will argue that algorithm performance and data annotation reliability are essentially the same concepts, and the same analytical tools can be used to investigate both. After this argument, I will wrap up the state-of-the-art methods, introduce the fundamental principle of variance decomposition, show how to estimate components using LMEMs, and define a meaningful reliability measure for data annotation and algorithm performance based on this decomposition. Finally, I will illustrate these techniques through an example analysis.

3.1 Untangling Terminology: Reliability, Agreement etc.

[Kri04] states the *measurement theortical conception of reliability* as follows:

> A research procedure is **reliable** when it responds to the same phenomena in the same way regardless of the circumstances of its implementation.

The terminological confusion starts when the terms "research procedure" and "circumstances of its implementation" are concretely interpreted. Krippendorff's main concern was to quantify the reliability of data annotations for a fixed sample of data points by a fixed selection of human coders — the "research procedure" — given their different response styles or exposition environments — the "circumstances of their implementation". Another interpretation that will be of interest in this chapter is to replace the "research procedure" with a machine learning model and the "circumstances of the implementation" as variability due to meta-parameter choices that are not specified by the algorithm but need to be made to train an algorithm on a data sample.

Reliability of measurements of nominal outcomes is frequently called *agreement* [Sho11; Hal12], and it is often reserved for the case of human raters representing the research procedure whose reliability is in question. However, in light of the above definition, the concept of *intra-rater agreement* (the consistency of annotation results of a human rater on repeated trials on same data) is essentially the same as *test-retest reliability* (the correlation between results of the same test on two occasions under otherwise identical circumstances and the concept of *inter-rater agreement* (the consistency of annotation results of two or more human raters on the same data) is the same as *test-test reliability* in measurement theory (the correlation between results of equivalent forms of tests performed under otherwise identical circumstances).

To add to the confusion, [Kri04] uses the term *stability* to denote intra-rater agreement and calls inter-rater agreement *reproducibility.* Further, the term *replicability* is introduced by [Dro+17] to denote consistency over different datasets from different domains or languages. [Ple18] lists different interpretations of the terms *replicability* and *reproducibility* and adds the term *repeatability*, which emphasizes the variable that different teams of researchers may add. Finally, [GFI16] clarify things a bit by stressing certain aspects within reproducibility by distinguishing *methods reproducibility*, *results reproducibility*, and *inferential reproducibility.*

In the following, I will stick to the term *reliability* and provide an operational definition that applies to data annotation and algorithm performance in NLP and data science alike, which is routed in a psychometric understanding of measurement.

3.2 A Unifying View on Algorithm Performance and Annotation

In this section, I want to establish this chapter's conceptual foundations. The reference points of these reflections will be the interrelated concepts of *measurement*, *score*, and *reliability* as established in psychometrics by [Bre01]:

Definition 3.1 (Measurement)**.**
A measurement is a non-random operation that assigns a numerical value to an object to acquire information about specific attributes or characteristics of that object.

Definition 3.2 (Score)**.**
The numerical value assigned to an object via a measurement is called the score of the object.

Ideally, the score is solely determined by the empirical property of the object, and other peculiarities of the measurement are of no significance. Hence, repeated measures of the same object would yield identical scores under all possible conditions. Unfortunately, this is rarely, if ever, the case. The extent to which replicated measurements produce similar scores is called *reliability*.

Definition 3.3 (Reliability)**.**
Reliability measures the degree of consistency in scores over replications of a measurement procedure (scoring procedure).

This definition implies several subtleties. Firstly, reliability is not a direct property of a measurement instrument but of scores obtained by this instrument under various conditions. Secondly, this definition makes no strict assertion about what constitutes a replication. This term is left open for the researcher to be defined [1].

In the case of data annotation, when human annotators annotate sentences, it is

[1] This doesn't mean that this choice is wholly arbitrary or that one can determine replications so that reliability is maximized.

somewhat likely that annotators yield non-equal annotations for a sentence. Thus it is necessary to investigate the reliability of the scores obtained for sentence scores replicated under varying human annotators or repeated annotations by the same annotator.

In the case of learning algorithm performance, one can think of a classifier obtained by the algorithm as a measurement. The meta-parameters of the algorithm, which need to be set to get a classifier, constitute a replication because varying them will yield potentially different classifiers[2].

Regardless of considering data annotation or learning algorithm performance, we have to distinguish between two cases:

- The output of the annotation or the classifier is immediately numeric or categorical, e.g., a sentiment annotation or a regression model.

- The output is not of this quality, e.g., a translation or a summary.

In the latter case, one can proceed as in the first case, when the output is mapped into a *performance score* via an evaluation metric. Regarding data annotation in interactive machine translation, possible performance evaluation metrics are post-editing time or human translation edit rate [Sno+06]. For machine learning predictions, for example, possible performance evaluation metrics in machine translation are BLEU [Pap+02] or TER [Sno+06]. Different cases of evaluation scores in the experimental examples are discussed below.

Before continuing my exposition, I want to discuss commonly used (descriptive) statistics proposed as reliability metrics.

3.3 State of the Art Methods for Annotation and Algorithm Performance Analysis

The most popular of such statistics are agreement coefficients such as Scott's π [Sco55], Cohen's κ [Coh60], or Krippendorff's α [Kri04] that are commonly used to measure reliability in data annotation processes in NLP and data science. I will present a principle discussion of these statistics, emphasizing some oddities and shortcomings that, while discussed in the literature [ZLD13], are not widely known in NLP and data science communities. Furthermore, I will discuss recently suggested bootstrap [ET93] inspired approaches to estimate the reliability of algorithm performance.

[2] This means output mappings of the classifiers are not identical for all inputs.

3.3.1 Agreement Coefficients for Data Annotation

Krippendorff's α coefficient is a widely used agreement measure in NLP, at least since the survey paper of [AP08]. Its attractiveness is founded in its conceptual simplicity –basically, the observed agreement is reduced by chance agreement, similar to Scott's π [Sco55] or Cohen's κ [Coh60]– and its applicability to multiple raters and all standard scales of measurements (nominal, ordinal, interval, and ratio variables). It can also be easily computed from experimental data by collecting relative count statistics instead of optimizing a machine learning model. This computational convenience results from the simple probability model used to calculate chance agreement. This model is an urn model where all observed ratings give the type and frequency of the balls, and the probabilities for observing a pair (or tuple) of observations are combinatorial.

As we will see, this model

- has the tendency to yield rather high values for chance agreement, and thus a low agreement when intuitively the opposite is the case.

- is susceptible to the actual observed values because minor variations in the observed ratings can result in dramatic differences between the corresponding α coefficient.

- results in an undefined agreement coefficient if no variation in the observed ratings exists.

I will illustrate these shortcomings by presenting exemplary computations of α for two raters and nominal ratings, using an example from [Kri04] for binary ratings of two raters A and B on 10 items:

	1	2	3	4	5	6	7	8	9	10
A	1	1	0	0	0	0	0	0	0	0
B	0	1	1	0	0	1	0	1	0	0

To compute the α coefficient, we first need to sum up the number of observed rating values in a matrix of two raters while omitting references to the individual raters. The entries of this matrix are called observed coincidences o_{ck}:

	0	1	
0	o_{00}	o_{01}	n_0
1	o_{10}	o_{11}	n_1
	n_0	n_1	n

	0	1	
0	10	4	14
1	4	2	6
	14	6	20

Second, we need to calculate expected coincidences e_{ck} to represent what could happen by chance. The following simple urn model can illustrate this random sampling process. Assume we write each rating of the two raters on a ball and put it in an urn. Then we draw two balls from the urn without replacing the first one. For our example, expected coincidences are computed as follows:

	0	1	
0	e_{00}	e_{01}	n_0
1	e_{10}	e_{11}	n_1
	n_0	n_1	n

	0	1	
0	9.6	4.4	14
1	4.4	1.6	6
	14	6	20

The expected coincidence e_{00} of chance agreement between raters A and B on two 0s is calculated by letting the first rater draw a 0 in 14 out of 20 cases and letting the second rater draw a 0 in $14-1$ out of $20-1$ cases. By multiplying these two probabilities by the total number of 20, we get the expected frequency of 9.6 pairs of two 0s. The remaining expected coincidences are computed accordingly, as shown below:

$$
\begin{aligned}
e_{00} &= \frac{n_0}{n} \cdot \frac{n_0 - 1}{n - 1} \cdot n = \frac{14}{20} \cdot \frac{13}{19} \cdot 20 = 9.6 \\
e_{11} &= \frac{n_1}{n} \cdot \frac{n_1 - 1}{n - 1} \cdot n = \frac{6}{20} \cdot \frac{5}{19} \cdot 20 = 1.6 \\
e_{01} &= \frac{n_0}{n} \cdot \frac{n_1}{n - 1} \cdot n = \frac{14}{20} \cdot \frac{6}{19} \cdot 20 = 4.4 \\
e_{10} &= \frac{n_1}{n} \cdot \frac{n_0}{n - 1} \cdot n = e_{01}
\end{aligned}
$$

From these coincidence tables, the α coefficient is computed as follows:

$$\begin{aligned}
\alpha &= \frac{\text{observed agreement - chance agreement}}{n - \text{chance agreement}} \\
&= 1 - \frac{\text{observed disagreement}}{\text{expected disagreement}} \\
&= 1 - \frac{o_{01} + o_{10}}{e_{01} + e_{10}} \\
&= 1 - \frac{o_{01}}{e_{01}}.
\end{aligned}$$

For the example above, this yields $\alpha = 1 - \frac{4}{4.421} = 0.095$. The idea of α as a measure of chance-corrected agreement is motivated by the values at the end of the range. An α value of 0, indicating the absence of reliability, is obtained when the observed disagreement is entirely due to chance and thus equal to the expected disagreement. An α value of 1, indicating perfect reliability, is obtained when there is no observed disagreement. In our example, α is relatively low at barely 10%, while the uncorrected observed agreement — the percent of cases of agreement out of all analyzed cases — is at 60%. The explanation for this discrepancy is that the assumed model of chance agreement attributes 56% of chance agreement, as can be seen by calculating $(9.6/20) + (1.6/20) = 56\%$.

Note that the definition of α above does not guarantee $\alpha \in [0, 1]$. For example, α will be negative for the following table of ratings:

	1	2	3	4	5	6	7	8
A	0	0	0	0	0	0	1	0
B	0	0	0	0	0	0	0	1

First, the uncorrected percent agreement is at $6/8 = 75\%$. The matrices of observed and expected coincidences are as follows:

	0	1	
0	12	2	14
1	2	0	2
	14	2	16

	0	1	
0	12.13	1.87	14
1	1.87	0.13	2
	14	2	16

The calculation of expected disagreement is again based on the value $e_{01} = e_{10}$:

$$\begin{aligned}
e_{00} &= (14/16)((14-1)/(16-1))16 = 12.13\\
e_{11} &= (2/16)((2-1)/(16-1))16 = 0.13\\
e_{01} &= (14/16)(2/(16-1))16 = 1.87
\end{aligned}$$

We note that the α value is negative since $\alpha = 1 - \frac{2}{1.87} = -0.07$. The explanation lies again in the computation of chance agreement which amounts to $(12.13/16) + (0.13/16) =$ 76.6%. This means that the observed agreement (75%) is poorer than chance.

Unfortunately, even if one agrees with the principle of maximum randomness, the stipulation of the chance agreement by a random sampling model has further ramifications. While values of α at the ends of the range were supposed to motivate the measure, extreme values can also be obtained by nonsensical abnormalities, defeating a clear interpretation of the measure. Consider the following table of binary ratings of two raters A and B on our 10 items:

	1	2	3	4	5	6	7	8
A	0	0	0	0	0	0	0	1
B	0	0	0	0	0	0	0	1

The uncorrected percent agreement amounts to 100%, and α reaches a maximum due to no observed disagreement: $\alpha = 1 - \frac{0}{e_{01}} = 1$. If this result is desired, consider a tiny change in the table that throws a wrench in the works:

	1	2	3	4	5	6	7	8
A	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	1

A change of one rating by one rater renders observed and expected coincidences equal:

	0	1	
0	14	1	15
1	1	0	1
	15	1	16

$$e_{00} = (15/16)((15-1)/(16-1))16 = 14$$
$$e_{11} = (1/16)((1-1)/(16-1))16 = 0$$
$$e_{01} = (15/16)(1/(16-1))16 = 1$$

This yields $\alpha = 1 - \frac{1}{1} = 0$, although the uncorrected percent agreement is still at $7/8 = 88\%$. Consider another tiny change in the table, yielding zero variation:

	1	2	3	4	5	6	7	8
A	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0

Now $\alpha = 1 - \frac{0}{0}$ is technically undefined. Nevertheless, [Kri04] arbitrarily defines it to be 0 in this case, while the uncorrected percent agreement is 100%.

Discussion. To summarize, chance-corrected agreement measures like Scott's π, Cohen's κ, or Krippendorff's α can be written in the following form:

$$\frac{\text{observed agreement - chance agreement}}{n - \text{chance agreement}}.$$

All measures stipulate a hypothetical model for chance agreement, where the central differences lie in choices such as sampling with replacement (Scott's π and Cohen's κ) or without replacement (Krippendorff's α) from distributions for individual raters (Cohen's κ) or for the observed ratings averaged over raters (Scott's π and Krippendorff's α).

A crucial similarity between the measures is the fact that the above-described counter-intuitive principle of maximum randomness, and the resulting abnormalities, apply to all chance-corrected agreement metrics in a similar way.[3] Furthermore, all listed shortcomings apply to all scales.

Arguably, the main shortcoming common to π, κ, and α is that these measures are descriptive statistics that do not permit concluding concrete raters and objects examined

[3] See [ZLD13] for an exhaustive list of paradoxes and abnormalities of chance-corrected agreement measures like π, κ, and α.

during an experiment. Further, agreement measures do not allow detailed analysis of the reason for high or low agreement by general properties of raters or meta-parameters for algorithms. However, a helpful measure should provide these possibilities.

3.3.2 Bootstrap Confidence Intervals for Model Evaluation

Inference beyond the observed samples of an experiment is indispensable for assessing the potential usefulness of an algorithm. In contrast to data annotation, where the focus is often reduced to particular human annotations used to generate a fixed and static dataset, algorithms should be used in a wide range of application contexts, not just for the specific experiment at hand. Thus, even if the interval scaled variant of Krippendorff's α would, in principle, be applicable to measure the reliability of model prediction processes, it does not make sense to estimate a single number indicating the reliability of a machine learning prediction for a given set of tested meta-parameters, without generalizing across the concrete meta-parameter settings and data that were used in a particular experiment.

It has been suggested in the machine learning community to address the problem of reliability of algorithm performance by computing confidence intervals for evaluation metrics computed on test data. In the following, I will look at the approaches of [Hen+18; Luc+18], who propose bootstrap-inspired resampling procedures to compute an interval estimator for evaluation scores on test data. The approach advocated in [Luc+18] aims to capture the variability of an evaluation metric introduced by a random search over meta-parameters during training and express it as an interval estimator for the expected maximum performance under a computational budget. [Hen+18] use bootstrap confidence intervals to compare the performance differences due to different meta-parameter choices in a reinforcement learning setting. The details of the implemented algorithms to construct bootstrap confidence sets vary from study to study. To keep the focus and stay aware of more information, I will briefly summarize the central concepts of confidence intervals and bootstrap techniques for their construction and sketch an algorithm to apply these ideas to construct confidence intervals for evaluation metrics with budget constraints.

Following [Sha03], confidence intervals are defined as:

Definition 3.4 (Confidence Interval).
Let $\mathscr{P}$ denote a family of distributions and $\theta \in \mathbb{R}$ be an unknown parameter of $P \in \mathscr{P}$. Further, let $\alpha \in (0,1)$ and $Y = (Y_1, Y_2, Y_3, ..., Y_n)$ be a random sample generated from the random process described by $\mathscr{P}$. Then the estimated interval $[\hat{\theta}_l(Y), \hat{\theta}_u(Y)]$ is called

a *confidence interval for θ at confidence level $1-\alpha$* if $\forall P \in \mathscr{P}$ holds

$$P(\hat{\theta}_l(Y) \leq \theta \leq \hat{\theta}_u(Y)) \geq 1-\alpha. \tag{3.1}$$

Formally, a confidence interval is a function of the randomly sampled data Y from which estimators of the lower bound $\hat{\theta}_l(Y)$ and the upper bound $\hat{\theta}_u(Y)$ need to be constructed. This construction must be done so that the actual parameter θ is covered by the interval by a fraction of at least $(1-\alpha)$ of all possible samples. The most prominent example of a confidence interval is the case of independent and identically distributed Gaussian data with an unknown mean. For this case, the bound estimators can be constructed analytically, yielding the well-known formula for a 95% confidence interval of the population mean μ, where $\bar{x}$ is the sample mean and $\sigma_{\bar{x}}$ denotes the standard error:

$$\bar{x} - 1.96\sigma_{\bar{x}} \leq \mu \leq \bar{x} + 1.96\sigma_{\bar{x}}. \tag{3.2}$$

Thus 95% of intervals constructed in this way on numerous samples of the same size will cover the population mean μ.[4]

In the case that the family $\mathscr{P}$ can not be specified for an application, confidence intervals can be constructed via nonparametric[5] bootstrap sampling distributions. A simple approach is the so-called *standard method*, which constructs bootstrapped confidence intervals by plugging bootstrap estimates of $\sigma_{\bar{x}}$ into Equation (3.2).[6]

A use case of particular interest to the machine learning community is the calculation of confidence intervals for the maximum out-of-sample performance of an evaluation metric under a given computational budget[7]. The variation to be quantified in these

[4] A realized confidence interval must not be interpreted in a probabilistic fashion: Once a sample is drawn and the confidence bounds are determined, the resulting interval either includes θ or not, but all involved quantities are non-random: $\hat{\theta}_l(Y)$ and $\hat{\theta}_u(Y)$ have been observed, and θ is an unknown, but non-random quantity. The $(1-\alpha)$ probability relates to the confidence of the estimation procedure, not to a specific calculated interval.

[5] The main principle of the nonparametric bootstrap is the substitution of the unknown data distribution by the empirical distribution obtained from the i.i.d data sample. Generating data from this distribution is equivalent to drawing with replacement from the original sample. This method is especially effective for large sample sizes.

[6] A method to construct a bootstrap confidence interval with better coverage of the true parameter is the so-called percentile method [ET93; Coh95]. In general, the construction of bootstrap confidence intervals is a somewhat delicate problem for which no general conclusive method has been found yet. Improved methods and an illustrative discussion of this topic are presented in [EH16].

[7] Corresponding to the general practice of choosing the best model out of all models obtained in a meta-parameter search

applications is due to potentially different choices of meta-parameter configurations that are visited during the search in model training. Let p_m denote a model trained under a meta-parameter configuration m, where M is the size of the set of all potential meta-parameter configurations, and $B \leq M$ is the computational *budget* that restricts the number of meta-parameter search trials. Following the key ideas presented in [Hen+18; Luc+18; Dod+19; Tan+20], a somewhat unconventional bootstrap-like procedure can be defined by resampling performance evaluation scores to compute a confidence interval for an evaluation metric under a computational budget:[8]

Algorithm 3.1 (Confidence Interval for Evaluation Metric under Computational Budget)

1. Generate M meta-parameter configurations for considered model class.
2. For each $m = 1, \ldots, M$: Train model p_m and calculate the performance evaluation score $u_m = u(p_m)$.
3. For each $B \leq M$: Construct a bootstrap distribution by K times drawing B random samples with replacement from $\{u_m \colon m = 1, \ldots, M\}$. For each sample, select the maximum performance score.
4. Calculate the mean and the standard deviation of this distribution. To construct a confidence interval plug both estimates into Equation (3.2).

The use of confidence interval for measuring the reliability of model prediction performance is two-fold: First, the confidence interval can be used to directly signify *error bars* that visualize the confidence bounds on the mean value in a plot. For example, Figure 3.1 shows the mean values as dots and 95% confidence intervals as vertical bars for the means of the evaluation metrics F1-score, precision, and recall for computational budgets (number of visited meta-parameter configurations) to train meta-parameter variants of Generative Adversarial Networks (GANs) [Luc+18]. Confidence bounds can then be used to assess the reliability of an evaluation under different meta-parameter settings. The rationale is that at the same level of confidence, smaller confidence bounds indicate less variability among the scores. Thus the maximum performance score obtained by the meta-parameter search is more likely to be repeatable.

Second, a bootstrap confidence interval can be used to perform a *conservative significance test*[9] by comparing confidence intervals. Given two mean evaluation scores of two

[8] Obviously, in this setup the probability to include the best performance metric $u^*_m = \max_{m=1,\ldots,M} u_m$ in the bootstrap sample is non-decreasing with B, this fact will result in smaller confidence intervals for larger B. Furthermore, note that this algorithm might give a somewhat misleading impression if the meta-parameter space of the algorithm is much larger than M.

[9] Conservative significance tests are characterized by the probability of incorrectly rejecting the null

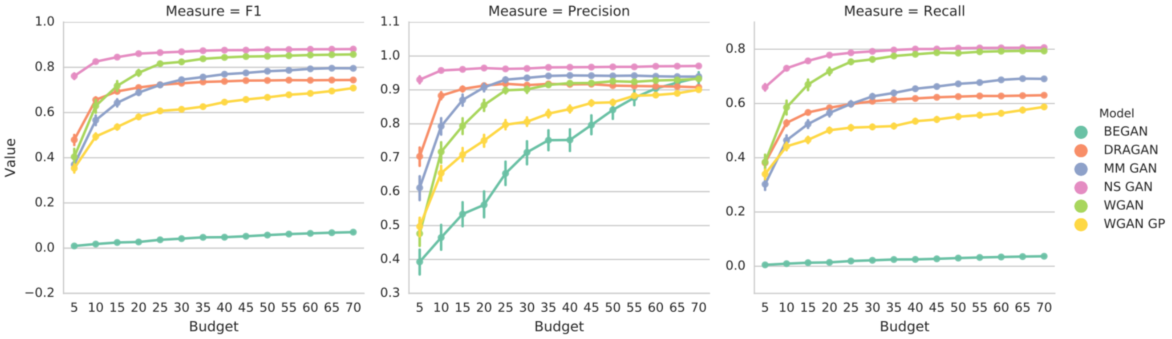

Figure 3.1: Mean and 95% confidence intervals for F1-score, precision, recall of GANs for different computational budgets. Graphics from [Luc+18].

competing systems and the confidence intervals about these means, one can prove that if the confidence intervals do not overlap — the upper bound of one is below the lower bound of the other — then the means will be significantly different [DS12].

Discussion. Bootstrap techniques are popular in NLP and data science since they can be applied to compute confidence intervals for complicated nonlinear evaluation metrics such as F1-score [MRS08], BLEU [Pap+02], or ROUGE [LH03], used for classification, machine translation, or summarization, respectively. The cited reason for the bootstrap's flexibility is that it does not make any assumptions about the underlying population distribution except that the original sample is representative of the population [Coh95]. However, to guarantee the correctness of a bootstrap confidence interval, a normality assumption on the sampling distribution of the evaluation metric has to be made, or else the evaluation metric u has to satisfy the condition of the existence of a monotone transformation $\hat{\phi} = g(u)$ such that the sampling distribution of $\hat{\phi}$ is normal. [ET93] list a few normalizing transformations whose existence guarantee the correctness of the bootstrap confidence interval, in the sense that the confidence bounds are the same when applying the bootstrap technique to the test statistic before and after transformation. Such correcting transformations are usually not considered when using bootstrap techniques for complex test statistics.

Another potential problem of bootstrap techniques is the failure of bootstrap consistency [Can+06]. Bootstrap inconsistency happens when the bootstrap distribution of a statistic doesn't converge to its actual distribution as the sample grows, leading

hypothesis being less than the nominal significance level. This tighter (but unknown) Type 1 error control results in a lower power compared to tests that operate at precisely the nominal level. More information on statistical significance testing is found in Chapter 2.

to a wrong approximation and false conclusion. Bootstrap inconsistency happens when the bootstrap data set is not representative. For 3.1, this can be the case when the M bootstrap configurations forming its data basis don't provide an accurate image of the actual meta-parameter space of the algorithm. Bootstrap inconsistency also occurs when the parameter to be estimated is on the boundary of the parameter space [And00; BF81], which is the case for bootstrap-inspired procedures aiming to provide inference for the maximum of the expected performance under a given budget [Dod+19; Luc+18].

An alternative to bootstrap methods is cross-validation techniques to compute confidence intervals for the expected performance scores. For example, [Die98] proposes 5 iterations of 2-fold cross-validation. In contrast, [NB99] propose cross-validation runs on several half-splits of the data separately to obtain conservative estimates of the standard error to construct standard confidence bounds. These methods can become quite computationally intensive since they involve several runs of training and evaluation on the obtained data splits and are thus often not feasible. Furthermore, as shown by [BG04], no unbiased estimator exists for cross-validation variance due to correlations among the evaluation scores (due to overlapping training sets) for each data split. Consequently, inference based on this estimator suffers from underestimating variance, leading to narrow standard confidence intervals with coverage below the nominal level. Only recently, [BHT23] introduced a nested cross-validation scheme to estimate standard errors more accurately, leading to confidence intervals with approximately correct coverage.

Lastly, and most importantly, neither expected maximum evaluation scores nor error bars based on confidence intervals allow us to assess the variation of algorithm performance in detail. A mean to this end is model-based approaches to reliability, as described in the next section.

3.4 Bridging the Gap: Model-based Reliability Analysis

In classical psychological measurement theory [LN68], an undifferentiated measurement error accompanies every experimental measurement. More recent work in psychometrics further investigates this error by employing variance component analysis to untangle multiple sources of variation that contribute to the variability in measurement [Bre01]. For data annotation, variance decomposition means decomposing the total variance into factors corresponding to measurement conditions such as raters, sentences, or interactions between raters and sentences. This idea can seamlessly be transferred to algorithm performance analysis, where the experimenter wants to investigate the influence of meta-parameters and input properties or interactions between these factors on algorithm performance variation.

In the subsequent section, I will introduce the central concepts of variance component analysis [SCM92], adapt these ideas to the fields of NLP and data science, and show how to utilize LMEMs [MS01] to estimate the variance components. Following this exposition, I will use variance components to define a reliability coefficient that quantifies the reliability of data annotation and model performance as the share of total variance attributable to differences between the measurement objects. Further, I will show the variance components can immediately be used to assess the contribution of individual meta-parameters to algorithm performance variation.

3.4.1 The Basic Principles of Variance Component Analysis (VCA)

Let us introduce the basic principles by considering a fictive example from interactive machine translation [Gre+14; Ben+16; SKR16; KSR18; KBR20]. This experiment's response $y = y_{sr}$ is an evaluation score measuring human annotation effort, e.g., human Translation Edit Rate [Sno+06], obtained for sentence s and rater r. The response can be rewritten by the following *tautological decomposition* as a sum of four components:

$$y_{sr} = \mu + (\mu_s - \mu) + (\mu_r - \mu) + (y_{sr} - \mu_s - \mu_r + \mu). \tag{3.3}$$

The components are

- the grand mean μ of the observed evaluation score across all potential raters r and all potential sentences s;
- the deviation $(\mu_r - \mu)$ of the mean score $\mu_r := \mathbb{E}_S\left[Y_{SR}|R = r\right]$ assigned by rater r from the grand mean μ;
- the deviation $(\mu_s - \mu)$ of the mean score $\mu_s := \mathbb{E}_R\left[Y_{SR}|S = s\right]$ assigned to sentence s from the grand mean μ;
- and the residuum, reflecting the deviation of the observed score y_{sr} from the sum of the first three.

Except for μ, each component varies between raters or sentences. Now, let us consider a probabilistic version of this tautology. Let Y_{SR} be a random variable obtained by independently sampling rater and sentence; thus $M_R := \mathbb{E}_S\left[Y_{SR}|R\right]$ and $M_S := \mathbb{E}_S\left[Y_{SR}|S\right]$ are also random[10]. Then $\mathbb{E}_{S,R}\left[Y_{SR}\right] = \mathbb{E}_S\left[M_S\right] = \mathbb{E}_R\left[M_R\right] = \mu$ and thus the expected value[11] of all deviations on the right-hand-side of (3.3) are 0. But more interesting than the first are the second moments of these terms. Obviously the covariance between $(M_R - \mu)$ and $(M_S - \mu)$ is 0 because rater and sentences are sampled independently. $(M_R - \mu)$ and $(M_S - \mu)$ are also uncorrelated with the residuum, as shown by the following calculation:

$$\begin{gathered}
\mathbb{E}_{S,R}\Big[(Y_{SR} - M_S - M_R + \mu)(M_S - \mu)\Big] = \\
\mathbb{E}_S\Big[\mathbb{E}_R\left[(Y_{SR} - M_S - M_R + \mu)(M_S - \mu)|S\right]\Big] = \\
\mathbb{E}_S\Big[\underbrace{\mathbb{E}_R\left[Y_{SR}|S\right]}_{=M_S} M_S - M_S^2 - \underbrace{\mathbb{E}_R\left[M_R|S\right]}_{=\mu} M_S + M_S\mu \\
- \underbrace{\mathbb{E}_R\left[Y_{SR}|S\right]}_{=M_S}\mu + M_S\mu + \underbrace{\mathbb{E}_R\left[M_R|S\right]}_{=\mu}\mu - \mu^2\Big] = 0
\end{gathered}$$

The first equality follows from the law of total expectation[12], and the second equality exploits the fact that raters and sentences are sampled independently. Since we have shown that all components are pairwise uncorrelated with each other, the total variance $\sigma^2(Y)$ can be decomposed into:

$$\sigma^2(Y) = \sigma_S^2 + \sigma_R^2 + \sigma_{residual}^2, \tag{3.4}$$

where σ_S^2 and σ_R^2 denote the variance due to sentences and raters, and $\sigma_{residual}^2$ denotes the residual variance component, including the variance due to the interaction of S and R.

In the psychometric approach to reliability of [Bre01], the conditions of measurement that contribute to variance in the measurement besides the objects of interest are called *facets* of measurement. In the example above, the objects of interest in our measurement are the sentences. They are the essential object to be measured. The only facet of measurement in this example is raters. An experiment based on this so-called one-facet fully crossed design would randomly select a finite subset of sentences and raters and observe the scores for all possible combinations. Adding additional facets, one would arrive at a so-called multi-facet design, which enables the explicit analysis of interaction

[10] Here I follow the convention to denote random variables with capital letters and their realization with small letters.

[11] All expectations are defined over both sentences and raters.

[12] $\mathbb{E}\Big[X\Big] = \mathbb{E}\Big[\mathbb{E}\left[X|Y\right]\Big]$ for a formal proof see [WHH05]

effects. For example, adding a facet, for instance, i of replicated annotation by the same rater on the same sentences would lead to the following two-facet fully crossed design:

$$\begin{aligned} y_{sri} = \mu &+ (\mu_s - \mu) + (\mu_r - \mu) + (\mu_i - \mu) \\ &+ (\mu_{sr} - \mu_s - \mu_r + \mu) \\ &+ (\mu_{si} - \mu_s - \mu_i + \mu) \\ &+ (\mu_{ri} - \mu_r - \mu_i + \mu) \\ &+ (Y_{sri} - \mu_{sr} - \mu_{si} - \mu_{ri} + \mu_r + \mu_s + \mu_i - \mu). \end{aligned} \tag{3.5}$$

Using the same techniques and arguments as above, one could establish the pairwise uncorrelatedness of these terms when sentences, raters, and instances are sampled independently. Consequently the total variance $\sigma^2(Y)$ can be decomposed into:

$$\sigma^2(Y) = \sigma_S^2 + \sigma_R^2 + \sigma_I^2 + \sigma_{SR}^2 + \sigma_{SI}^2 + \sigma_{RI}^2 + \sigma_{residual}^2. \tag{3.6}$$

The facets of measurement in this design include raters R, instance I, and facets for interactions SR, SI, and RI, with objects of measurement being sentences S.

Estimating the variance components has traditionally been done by ANOVA estimators based on expected mean square equations. These date back to [Fis25] and are discussed extensively in [Bre01]. A more flexible alternative is to model variance components as random effects in LMEMs (2.2 for more details). One can immediately see that the probabilistic version of (3.3) is an LMEM just by applying familiar notation to it:

$$Y_{SR} = \underbrace{\mu}_{=:\beta_0} + \underbrace{(M_S - \mu)}_{=:b_S} + \underbrace{(M_R - \mu)}_{=:b_R} + \underbrace{(Y_{SR} - \mu_S - \mu_R + \mu)}_{=:\epsilon}. \tag{3.7}$$

where

$$\mathbf{b} = \begin{bmatrix} b_R \\ b_S \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_R^2 & 0 \\ 0 & \sigma_S^2 \end{bmatrix}),$$
$$\epsilon \sim \mathcal{N}(0, \sigma_{residual}^2).$$

β_0 denotes the intercept and is the only fixed effect of the model. b_S and b_R are random effects corresponding to sentences and raters. Such a model is called a *random effects only model* in the LMEM literature. Nevertheless, we can use the same estimation techniques to get estimates of σ_R^2, σ_S^2, and $\sigma_{residual}^2$.

3.4.2 VCA based Reliability Coefficients

The final step to a model-based approach to reliability is the definition of a coefficient that puts the important variance component in relation instead of inspecting it in isolation. The key concept is the *intra-class correlation coefficient (ICC)*, dating back to [Fis25]. A fundamental interpretation of the ICC is as a measure of the proportion of variance attributable to the objects of measurement. The name of the coefficient is derived from the goal of measuring how strongly objects in the same class are related. The coefficient is computed as the variance ratio between objects of interest σ^2_B to the total variance σ^2_{total}. The latter includes variance within objects of interest σ^2_W, or simply undifferentiated residual variance σ^2_ϵ:

$$ICC = \frac{\sigma^2_B}{\sigma^2_{total}} = \frac{\sigma^2_B}{\sigma^2_B + \sigma^2_W} = \frac{\sigma^2_B}{\sigma^2_B + \sigma^2_\epsilon}. \tag{3.8}$$

For instance, assume that the measurement objects are machine-translated sentences in machine translation human-based quality judgments. An annotation is considered reliable if most of the variance observed among annotations is explained by variance between sentences and not by variance between raters (within sentences), caused by inconsistencies of human annotators, or by residual variance due to unaccounted facets of the measurement procedure. Hence, variance attributable to the measurement objects, here sentences, should dominate the decomposition.

To quantify the components σ^2_B and $\sigma^2_W = \sigma^2_\epsilon$ from the obtained experimental data, we fit an appropriate version of an LMEM to estimate the variance due to the objects of interest and optionally for various facets. In the context of data annotation, Brennan's [Bre01] approach is first to estimate variance components from initial experimental observations and then to use these estimates to optimize the design of the measurement procedure further for final use.[13] The primary optimization technique is to "average over facets", e.g., instead of assigning the quality rating of a single annotator to a machine-translated scenting, one assigns the average rating of n_r raters as the quality score. The second step of Brennan's workflow is primarily interesting for data annotation and only of minor importance to our more critical use-case of algorithm performance analysis. Nevertheless, when we adapt Brennan's ideas to algorithm performance analysis, we will

[13] [Bre01] calls the first a generalization study (or G-study) associated with a universe of admissible observations, and the second a decision study (or D-study) associated with a universe of generalization. We will not use this terminology to avoid confusion with the terms "generalization" and "decision" in machine learning.

remember that $n_X = 1$ for all X in every presented formula.

Let us consider the two-facet fully crossed design with the objects of interest being sentences s and measurement facets for rater r and instance i. Furthermore, let n_r denote the number of raters and n_i the number of instances. [Bre01] interprets total variance as the variance between objects of interest, here σ_s^2, plus the absolute error variance σ_Δ^2 that includes variance components for all facets and interactions, except σ_s^2:

$$\sigma_\Delta^2 = \frac{\sigma_r^2}{n_r} + \frac{\sigma_i^2}{n_i} + \frac{\sigma_{sr}^2}{n_r} + \frac{\sigma_{si}^2}{n_i} + \frac{\sigma_{ri}^2}{n_r n_i} + \frac{\sigma_{residual}^2}{n_r n_i}. \tag{3.9}$$

[Bre01] then defines an absolute reliability coefficient Φ[14] that relates the variance between objects of interest σ_s^2 to the total variance:

$$\Phi = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_\Delta^2}. \tag{3.10}$$

In the algorithm performance analysis where $n_X = 1$ for all X, the denominator of (3.10) equals the total variance of the performance metric. Thus now we can define the following reliability coefficient for algorithm performance:

Definition 3.5 (Algorithm Performance Reliability Coefficient).
Assume meta-parameters $h_1, h_2, \ldots, h_H$ and selected interactions between meta-parameters $h_i h_j$. Then we call φ, computed by the ratio of substantial variance attributable to inputs σ_s^2 to the total variance of evaluation metric $\sigma^2(Y)$, the performance reliability of an algorithm.

$$\varphi := \frac{\sigma_s^2}{\sigma^2(Y)} = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_\Delta^2} \quad \text{where} \quad \sigma_\Delta^2 = \sum \sigma_{f_i}^2 + \sum \sigma_{h_i h_j}^2 + \sigma_{residual}^2.$$

A desired property of φ is its invariance versus the LMEM used to estimate σ_S^2 from the evaluation data collected during a machine learning experiment. I will show that this

[14] [Bre01] calls this coefficient the "index of dependability". We will not use this naming in the following. He also introduces a relative reliability coefficient that is based on a relative error variance $\sigma_\delta^2 = \frac{\sigma_{sr}^2}{n_r} + \frac{\sigma_{si}^2}{n_i} + \frac{\sigma_{residual}^2}{n_r n_i}$ that only sums up variance components interacting with the items of interest. [Bre01] denotes this coefficient by $\mathbb{E}\rho^2$ and calls it "generalizability coefficient". The relative reliability coefficient focuses on the stability of the relative ordering of objects of interest. In contrast, the absolute reliability coefficient focuses on the homogeneity of absolute performance scores for objects of interest across measurement instances. In the experiments presented in this chapter, we will focus on the absolute reliability coefficient.

is always the case because the evaluation data is an instance of a partially crossed experimental design, which is sufficient to guarantee that σ_s^2 can be estimated independently of all other meta-parameter variance components. Thus it doesn't matter how detailed the variance of an experiment is decomposed, the estimated number for σ_S^2 and thus φ will be the same.

I will formalize the argument following the notation and parlance of [SCM92]. In this framework, variables that we called facets so far are termed *factors*, and their manifestations are called *levels*. Typically factors are used to specify a systematic structure in the data. The extent to which different levels of a factor affect the dependent variable is called the *effect* of a level of a factor on the response. *Effect estimators* are often obtained via the maximum likelihood principle. Let us focus on the effect estimator σ_S^2. To describe the structure of the data obtained by an experiment, we define the following terms:

Definition 3.6 (Crossed and Balanced Design)**.**
A factor is called crossed if all its levels are observed with all combinations of factor levels of the other factors realized in the experiment An experimental design is called partially crossed if at least one factor is crossed and fully crossed if all factors are crossed. An experimental design is balanced if the number of observations is the same for all factor combinations realized in an experiment.

Now, let us take a step back and looking how the evaluation data is generated. The experimenter obtains a collection of models by retraining an algorithm following his meta-parameter grid; then, he evaluates all the models on the test set. Thus each test set input is evaluated by all models and hence by all meta-parameter level combinations. Therefore, the factor test set input S is crossed. The design is balanced because every model (equivalent to meta-parameter configuration) is applied to all test set inputs.

The following proposition establishes the desired property.

Proposition 3.1 (Invariance of φ.)**.**
The variance component of the factor test set input σ_S^2 can be estimated irrespectively of all other meta-parameter variance components used to analyze the observed evaluation data variance obtained of an machine learning experiment.

Proof. In order to proof this assertion, we will derive the sum-of-squares based variance component estimators for the special case of a two factor fully crossed balanced design[15]. In contrast to the maximum likelihood estimator introduced in 2.2 this estimator has an analytic solution allowing us to immediately verify our claim. Because both estimators are consistent[16] under the discussed model, the result also holds asymptotically for maximum likelihood estimator.

We start our argument by restating the discussed model using the parlance introduced in this section:

$$Y_{SH} = \beta_0 + b_S + b_H + \epsilon_{SR} \tag{3.11}$$

where

$$\mathbf{b} = \begin{bmatrix} b_R \\ b_S \\ \epsilon \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} , \begin{bmatrix} \sigma_S^2 & 0 & 0 \\ 0 & \sigma_H^2 & 0 \\ 0 & 0 & \sigma_{residual}^2 \end{bmatrix})$$

and y_{sh} denotes the observed evaluation metric for the test sentence $s = 1, \dots, n$ and the model trained with hyper-parameter value $h = 1, \dots, m$. We further assume that the n sentences and m hyper-parameter values are independent samples from all possible appropriate test sentences and hyper-parameter values. Because we evaluate every test sentence for every model we realize a two factor fully crossed balanced design, allowing us to decompose the total sum-of-squares (SST) into the three components SSS (depends only on s), SSH (depends only on h) and the residual sum-of-squares (SSR):

$$\underbrace{\sum_{s=1}^{n}\sum_{h=1}^{m}(y_{sh} - \bar{y}_{..})^2}_{=:SST} = \underbrace{m\sum_{s=1}^{n}(\bar{y}_{s.} - \bar{y}_{..})^2}_{=:SSS} + \underbrace{n\sum_{h=1}^{m}(\bar{y}_{.h} - \bar{y}_{..})^2}_{=:SSH} + \underbrace{\sum_{s=1}^{n}\sum_{h=1}^{m}(y_{sh} - \bar{y}_{s.} - \bar{y}_{.h} + \bar{y}_{..})^2}_{=:SSR}$$

[15] This limitation is only for convenience and doesn't narrow the substance of the proof. An extension of the presented argument to larger designs can be found in [SCM92].

[16] Consistency is a general property of maximum likelihood estimators (see for instance [Vaa98]). Sum-of-Squares are essentially second moments of random variables. Under mild technical restrictions satisfied by the discussed model the Law of large numbers applies to them.

where

$$\bar{y}_{..} := \frac{1}{nm}\sum_{s=1}^{n}\sum_{h=1}^{h} y_{sh} \overset{3.11}{=} \beta_0 + \underbrace{\frac{1}{n}\sum_{s=1}^{n} b_s}_{=:\bar{S}} + \underbrace{\frac{1}{m}\sum_{h=1}^{m} b_h}_{=:\bar{H}} + \underbrace{\frac{1}{nm}\sum_{s=1}^{n}\sum_{h=1}^{m} \epsilon_{sh}}_{=:\bar{\epsilon}_{..}}$$

$$\bar{y}_{s.} := \frac{1}{m}\sum_{h=1}^{m} y_{sh} \overset{3.11}{=} \beta_0 + b_s + \bar{H} + \underbrace{\frac{1}{m}\sum_{h=1}^{m} \epsilon_{sh}}_{=:\bar{\epsilon}_{s.}}$$

$$\bar{y}_{.h} := \frac{1}{n}\sum_{s=1}^{n} y_{sh} \overset{3.11}{=} \beta_0 + \bar{S} + b_h + \underbrace{\frac{1}{n}\sum_{s=1}^{n} \epsilon_{sh}}_{=:\bar{\epsilon}_{.h}}$$

Using this equalities SSS, SSH and SSR can be expressed as:

$$SSS = m\sum_{s=1}^{n}\left[(b_s - \bar{S}) + (\bar{\epsilon}_{s.} - \bar{\epsilon}_{..})\right]^2$$

$$SSH = n\sum_{h=1}^{m}\left[(b_h - \bar{H}) + (\bar{\epsilon}_{.h} - \bar{\epsilon}_{..})\right]^2$$

$$SSR = \sum_{s=1}^{n}\sum_{h=1}^{m}(y_{sh} - \bar{y}_{s.} - \bar{y}_{.h} + \bar{y}_{..})^2$$

Let us now focus on SSS and calculate its expectation:

$$\mathbb{E}\left[SSS\right] = m\sum_{s=1}^{n}\mathbb{E}\left[[\underbrace{(b_s - \bar{S})}_{=:b_s^c} + \underbrace{(\bar{\epsilon}_{s.} - \bar{\epsilon}_{..})}_{=:\bar{\epsilon}_{s.}^c}]^2\right]$$

By assumption $\mathbb{E}[b_s^c] = 0$, $\mathbb{E}[\bar{\epsilon}_{s.}^c] = 0$ and b_s^c and $\bar{\epsilon}_{s.}^c$ are stochastically independent, therefore:

$$\mathbb{E}\left[SSS\right] = m\sum_{s=1}^{n}\mathbb{V}\left[b_s^c + \bar{\epsilon}_{s.}^c\right] = m\sum_{s=1}^{n}\mathbb{V}\left[b_s^c\right] + \mathbb{V}\left[\bar{\epsilon}_{s.}^c\right]$$

By assumption b_s and $\epsilon_s h$ are both sampled iid, hence:

$$\mathbb{V}\left[b_s^c\right] = \mathbb{V}\left[b_s\right] + \mathbb{V}\left[\bar{S}\right] - 2\mathbb{C}\left[b_s, \bar{S}\right] = \frac{n-1}{n}\sigma_S^2$$

$$\mathbb{V}\left[\bar{\epsilon}_{s.}^c\right] = \mathbb{V}\left[\bar{\epsilon}_{s.}\right] + \mathbb{V}\left[\bar{\epsilon}_{..}\right] - 2\mathbb{C}\left[\bar{\epsilon}_{s.}, \bar{\epsilon}_{..}\right] = \frac{n-1}{nm}\sigma_{residual}^2$$

So that, we can finally see that

$$\mathbb{E}\left[SSS\right] = (n-1)(m\sigma_S^2 + \sigma_{residual}^2).$$

Similar calculations for the expectations of SSH and SSR yield the following results:

$$\mathbb{E}\left[SSH\right] = (m-1)(n\sigma_H^2 + \sigma_{residual}^2)$$
$$\mathbb{E}\left[SSR\right] = (m-1)(n-1)\sigma_{residual}^2$$

Replacing the expectation with the observed sum-of-squares yields the sum-of-squares estimators for the variance components σ_S^2, σ_H^2 and $\sigma_{residual}^2$. Obviously, the right hand side of

$$\hat{\sigma}_S^2 = \frac{1}{m}\left(\frac{SSS}{n-1} - \frac{SSR}{(m-1)(n-1)}\right)$$

contains no reference of $\hat{\sigma}_H^2$.□

In the following, I will illustrate applying the suggested techniques for data annotation and algorithm performance. In the former case, I will analyze the reliability of machine translation quality judgments obtained from human annotators. Facets of measurement are human raters and repeated score instances of the same sentence and annotator.

In the second case, objects of measurement are test sentences to be translated by an algorithmically generated machine translation system. The score is an automatic evaluation metric (TER) comparing model outputs against gold standard references. Facets of measurement are meta-parameters of the algorithm used to generate the machine translation systems, properties of test sentences, and interactions between these factors.

The data for both examples were generated during a study conducted by [KBR20]. This study aimed to improve the performance of a pre-trained neural machine translation system by using human translation quality judgments as supervision signals in fine-tuning. They investigated two modes of human feedback. In the first mode, "Marking", human raters mark erroneous words in the machine translation output (hypothesis) using an annotation interface to highlight them. In the second mode, called "Post Edit", human raters correct hypothesis by deleting, inserting, and replacing words or parts of words. Nevertheless, the fraction of changed tokens of a hypothesis was calculated, and this correction rate was used to quantify the translation quality. Thus a lower score reflects a better translation. Each of the ten human raters annotated a subset of five example sentences three times to assess the reliability of data annotation. We will study the reliability of user feedback on these data. For reasons of efficiency and cost, the data annotation for the subsequent fine-tuning process –repeated several times with different meta-parameter configurations– was designed so that every hypothesis was annotated by another user, and no user saw the same hypothesis twice.

3.4.3 Data Annotation Analysis

A first impression of the reliability of the annotation procedure used by [KBR20] is given by Figure 3.2. We see that the same hypothesis tendentially receives a greater quality score for "Post Edit" annotations than for "Marking" annotations from the same rater and that the ratings are very homogeneous for most raters and hypotheses. It is worth noting that the spread of quality scores assigned to the sentences is much larger for "Post Edit" than for "Marking" based judgments. Let us now conduct a model-based reliability study for these data. The experiment design is a two-facet fully crossed design with a variance component for sentences s (hypothesis), raters r, instantiations i, and interactions sr, si, and ri. To analyze the obtained data from this experiment, we decompose the variance of the quality judgments Y according to the following model:

$$Y = \beta_0 + b_s + b_r + b_i + b_{sr} + b_{si} + b_{ri} + \epsilon, \tag{3.12}$$

Where $\mu = \beta_0$ is the intercept (grand mean) and

$$\begin{bmatrix} b_s \\ b_r \\ b_i \\ b_{sr} \\ b_{si} \\ b_{ri} \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} , \begin{bmatrix} \sigma_s^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_r^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_i^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{sr}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{si}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{ri}^2 \end{bmatrix}),$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2_{residual}).$$

The estimated variance components for "Marking" based quality scores are presented in Table 3.1. The reliability score $\phi = 12\%$ (shown in the first line percent column) and

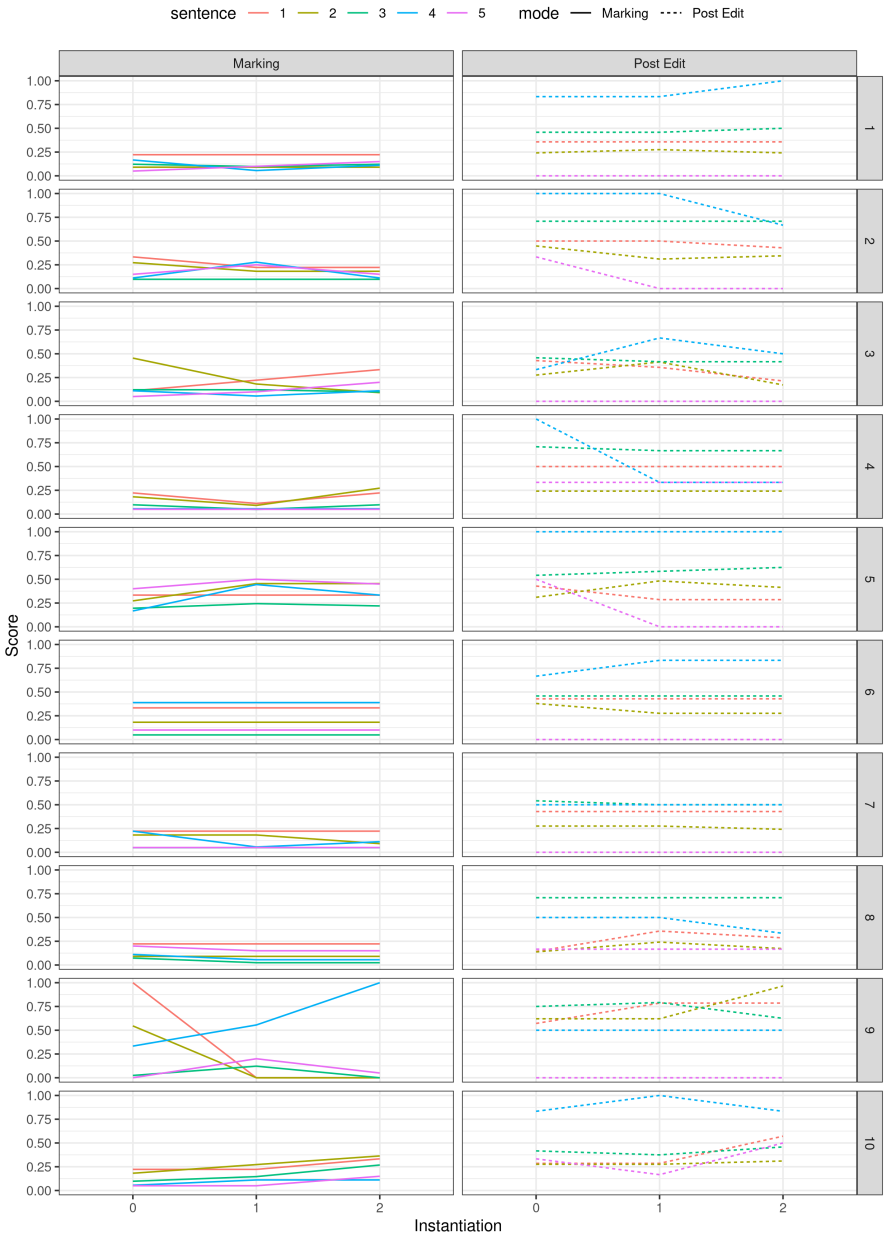

Figure 3.2: Sentence-wise quality judgment score computed as the ratio of marked or edited words per sentence, for 3 rating instantiations of 5 sentences by each of 10 raters.

pictures that only 12% of the variance can be attributed to variations between sentences (objects of measurement) and an overwhelming 88% must be attributed to nuisance factors of the procedure. The decomposition shows that a large fraction $\sigma_r^2 = 14.2\%$ of measurement variation can be attributed to different marking styles of raters, and style differences are not uniform across sentences, as seen by $\sigma_{rs}^2 = 16.1\%$. The contribution

Table 3.1: Variance components in translation marking experiment.

Component	**Variance**	**Percent**
sentence s	0.0030	**12.0**
rater r	0.0036	14.2
instantiation i	0.0000	0.0
interaction sr	0.0041	16.1
interaction si	0.0000	0.0
interaction ri	0.0000	0.0
residual	0.0145	57.6

Table 3.2: Variance components in translation post-editing experiment.

Component	**Variance**	**Percent**
sentence s	0.0479	**60.4**
rater r	0.0014	1.7
instantiation i	0.0000	0.0
interaction sr	0.0187	23.7
interaction si	0.0000	0.0
interaction ri	0.0006	0.8
residual	0.0106	13.4

of all other components is negligible. Thus a large amount of variance $\sigma^2_{residual} = 57.6\%$ can not be explained.

The variance component analysis for the "Post Edit" based quality judgments shown in Table 3.2 yields an entirely different picture. The reliability is approximately five times larger $\phi = 60.4\%$ than for "Marking", and σ^2_s is more significant than any other component. We also see that the non-substantial fraction of variance mostly comprises two components, namely $\sigma^2_{sr} = 23.7\%$ and $\sigma^2_{residual} = 13.4\%$.

Based on the experimental findings, post-edit quality judgments are more reliable than marking-based ones. But, according to the guidelines of [KL16], both procedures fail to yield φ values between 75% and 90%, which qualifies as good reliability.

Besides just providing insights into the peculiarities of particular facets of measurement, the above decomposition can also serve as a basis for designing an improved but still implementable procedure. In concreto, we can reduce the variation attributable to a facet

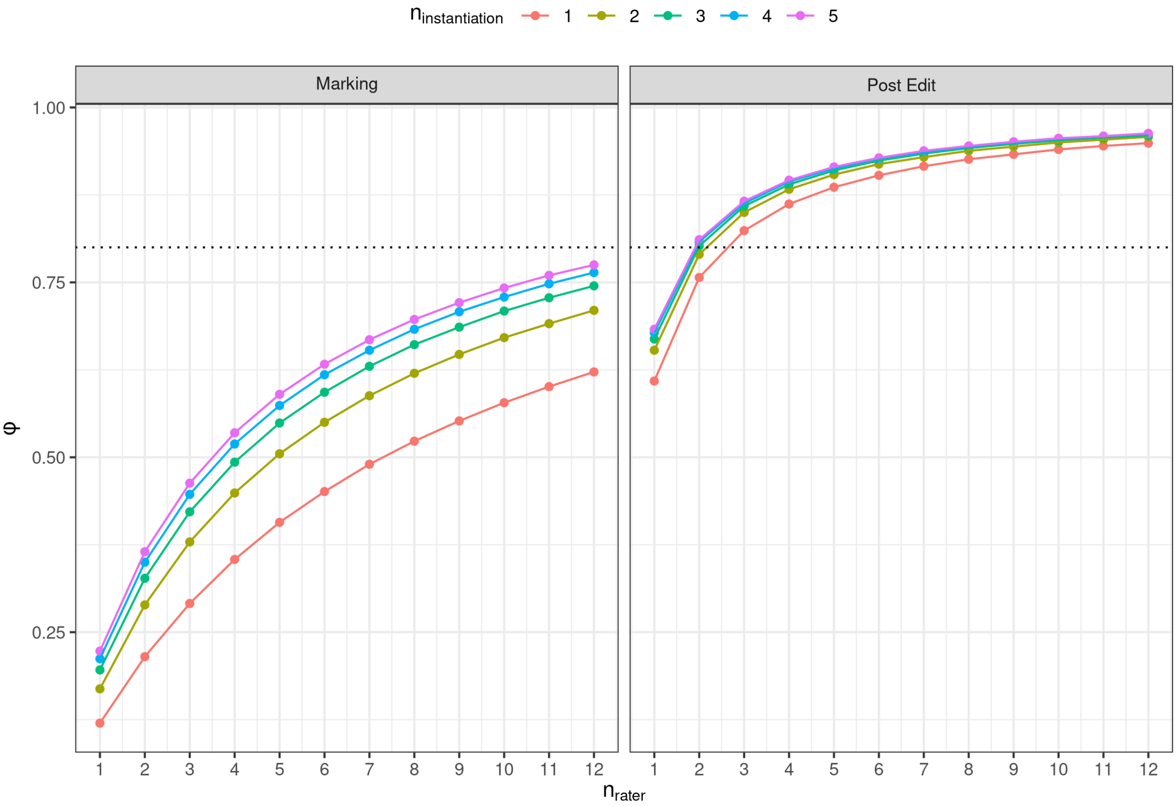

Figure 3.3: Reliability coefficient φ for data annotation performance in "Marking" or "Post Edit" mode, generalized to 5 rating instantiations and 12 raters.

f by averaging over n_f repeated measurements of the same object for different instances of the facet. This fact is expressed in (3.9).

To find the best trade-off between implementation cost and reliability, we compute the reliability coefficients for a grid of feasible implementations. Figure 3.3 shows the reliability coefficients for quality judgments obtained when averaged over combinations of $n_i = 1, .., 5$ instantiations and $n_r = 1, .., 12$ raters.

This study shows that the reliability of post-edit-based quality judgments can easily be pushed to a satisfying level when we average quality judgments over 2 to 3 raters to obtain the final score for a sentence and that it would require a tremendous and nearly impracticable effort to do so for marking based quality ratings.

3.4.4 Algorithm Performance Analysis

Typical machine learning experiment evaluation reports state nothing more than a single value of a metric for an algorithm. This score is usually found by selecting the best algo-

Table 3.3: Meta-parameters values used in the basic fine-tuning experiment of neural machine translation model on human marking data in [KBR20]. For an extended grid search, values in **bold face** are added.

Meta-parameter	Grid values			
learning_rate	0.0001	0.0003	0.0005	**0.001**
random_seed	42	43	44	
encoder_dropout	**0**	0.2	0.4	0.6
decoder_dropout	**0**	0.2	0.4	**0.6**
decoder_dropout_hidden	**0**	0.2	0.4	0.6
delta_scheme	(-0.5:0.5)	(0:1)		

rithm instance obtained by an extensive meta-parameter search (see [SGM19; Dod+19; Tan+20; Hen+18; Luc+18] for a discussion). Such a practice confuses the concept of an algorithm with an algorithm instance and discards a lot of information about an algorithm created during the experiment. It is also insufficient for judging the performance of an algorithm since it provides no insights about the performance homogeneity under differing meta-parameter settings. In this section, I will show how the techniques introduced in (2.2) can be used to:

- Quantify the performance homogeneity of an algorithm using the concept of reliability.
- Assess the influence of meta-parameters on the performance homogeneity of an algorithm.

I will also discuss how the results of these assessments change depending on the meta-parameter grid guiding the meta-parameter search. Further, I will show how LMEMs can incorporate data characteristics into this analysis. Such information is handy for machine learning practitioners who can use it to reduce the size and time spent on meta-parameter searches based on the characteristic of their data at hand.

As mentioned before, I will use the experiments on interactive machine translation by [KBR20] to illustrate an algorithm performance analysis. For recapitulation, this study aimed to use human markings and post-edits to improve a neural machine translation system. To this end [KBR20] fine-tuned a baseline system[17] based on this feedback.

[17] This system used is a encoder-decoder recurrent neural networks (RNNs) with attention [LPM15; BCB15], 4 bi-directional encoder and 4 decoder layers with 1,024 units each, and embedding layers of size 512 pre-trained on over 6 million parallel sentences, and fine-tuned on another 1,042 sentences.

The following presentation will be limited to marking-based annotations. This algorithm includes the following meta-parameters: Values of initial learning rate (`learning_rate`), seed of random number generator (`random_seed`), probability of zeroing out hidden connections during training of encoder (`encoder_dropout`), decoder (`decoder_dropout`), and hidden layers of the decoder (`decoder_dropout_hidden`) and the scheme to weight positive and negative markings in the training objective (`delta_scheme`). Ranges of meta-parameter values are shown in Table 3.3. In their original study, Kreutzer et al. considered only a subset of 27 of all possible combinations. I then trained the models for the missing meta-parameter combinations to show how analysis results would differ between a partial and a fully crossed design. I extended the grid to show the effect of incorporating extremer meta-parameter values. I call these three meta-parameter grids:

- *partial grid* consisting of the original 27 instances trained by Kreutzer et al.
- *full grid* consisting of 324 instances for a fully crossed original grid
- *extended grid* consisting of 1536 instances for a fully crossed extended meta-parameter grid

The evaluation set consists of 1,041 test sentences[18](the objects of interest). The evaluation data was generated by generating translations for this sentence using all models and calculating the TER evaluation score [Sno+06] against references for each translation.

Algorithm Performance Reliability and Meta-Parameter Importance

To obtain a variance decomposition that allows us to calculate φ and to assess the importance of a meta-parameter, we have to specify a random-effects-only LMEM of the following form:

[18] Note that two of the 1,043 test sentences reported in [KBR20] were duplicates that we removed in our LMEM experiments.

$$
\begin{aligned}
TER = {} & \beta_0 \\
& + b_s \\
& + b_{learning\_rate} \\
& + b_{random\_seed} \\
& + b_{encoder\_dropout} \\
& + b_{decoder\_dropout} \\
& + b_{decoder\_dropout\_hidden} \\
& + b_{delta\_scheme} \\
& + \epsilon
\end{aligned}
\tag{3.13}
$$

where

$$
\begin{bmatrix} b_s \\ b_{learning\_rate} \\ b_{random\_seed} \\ b_{encoder\_dropout} \\ b_{decoder\_dropout} \\ b_{decoder\_dropout\_hidden} \\ b_{delta\_scheme} \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} , \begin{bmatrix} \sigma_s^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{l\_r}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{r\_s}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{e\_d}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{d\_d}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{d\_d\_h}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{d\_s}^2 \end{bmatrix}),
$$

$$
\epsilon \sim \mathcal{N}(0, \sigma_{residual}^2).
$$

Applying this model to the evaluation data for the partial grid yields results presented in Table 3.4, the full grid yields results shown in Table 3.5, and the extended grid yields results displayed in Table 3.6.

Comparing the partial and full grid results, we see that the estimated variance components for sentences are numerically nearly identical in both cases $\varphi > .9$, indicating a very homogeneous algorithm performance. A detailed look at the other variance components shows that no individual meta-parameter dramatically influences the performance in both cases. This finding suggests that Kreutzer et al. were able to use their accumulated knowledge about the model architecture to steer the search towards instances with high variation because all the additional samples of the full grid show similar performance to

Table 3.4: Variance components in *partial meta-parameter grid* search for the basic fine-tuning experiment of neural machine translation model on human marking data in [KBR20].

Component	**Variance**	**Percent**
sentence s	0.0580	**91.6**
residual	0.0053	8.4
learning_rate	0.0008	0.0
decoder_dropout_hidden	0.0000	0.0
encoder_dropout	0.0000	0.0
random_seed	0.0000	0.0
decoder_dropout	0.0000	0.0
delta_scheme	0.0000	0.0

Table 3.5: Variance components in *full meta-parameter grid* search for the basic fine-tuning experiment of neural machine translation model on human marking data in [KBR20].

Component	**Variance**	**Percent**
sentence s	0.0584	**91.7**
residual	0.0053	8.3
learning_rate	0.0000	0.0
encoder_dropout	0.0000	0.0
decoder_dropout	0.0000	0.0
decoder_dropout_hidden	0.0000	0.0
random_seed	0.0000	0.0
delta_scheme	0.0000	0.0

instances already in the partial grid.

For the extended grid, we stretched the dropout values to 0, effectively turning off the regularization effect of dropout and adding a rather large learning rate, possibly introducing instability in training. Thus we expect a variance component analysis on this extended grid to result in a lower reliability coefficient due to more heterogeneous model performance, which should increase the fraction of non-sentence-related variance. Indeed as shown in Table 3.6, we see that the variance corresponding to learning rate and residual variance increase, while substantial variance slightly decreases, resulting in

Table 3.6: Variance components in *extended meta-parameter grid* search for the basic fine-tuning experiment of neural machine translation model on human marking data in [KBR20].

Component	**Variance**	**Percent**
sentence s	0.0575	**88.4**
residual	0.0074	11.3
learning_rate	0.0001	0.2
decoder_dropout	0.0000	0.0
encoder_dropout	0.0000	0.0
decoder_dropout_hidden	0.0000	0.0
random_seed	0.0000	0.0
delta_scheme	0.0000	0.0

a lower but still good reliability of φ of 88.4%.

This example taught us that an experience-informed search over a partial meta-parameter grid could yield a reasonably accurate picture of an algorithm's performance reliability. Nevertheless, omitting extreme meta-parameter values can cause a positively biased estimate. Therefore I recommend that either the experimenter (often identical to the algorithm creator) explicitly rule out these values in the definition of the algorithm or include them in his (partial) grid.

If one is only interested in the determination of φ, an analysis based on the simpler model:

$$TER = \beta_0 + b_s + \epsilon \tag{3.14}$$

where

$$b_s \sim \mathcal{N}(0, \sigma_s^2) \quad \text{and} \quad \epsilon \sim \mathcal{N}(0, \sigma_{residual}^2)$$

will yield identical estimates of φ due to Proposition 3.1 (see Table 3.7) for empirical confirmation). The technical benefits of using this model are faster convergence and increased numerical stability.

The quantification of meta-parameter importance by ANOVA-type techniques was previously suggested by [HHL14; ZLH20]. I show how variance component analysis based

Table 3.7: Variance components estimated using the minimal LMEM (3.14).

Grid	Component	Variance	Percent
partial	**sentence** s	0.0580	**91.6**
	residual	0.0054	8.4
full	**sentence** s	0.0584	**91.7**
	residual	0.0053	8.3
extended	**sentence** s	0.0575	**88.4**
	residual	0.0075	11.6

on the random effects of an LMEM can be used for this purpose. This technique applies to the data obtained for a wide range of meta-parameter search heuristics, e.g., the standard practice of a meta-parameter search over an incomplete configuration space, either guided by the experience of the modeler (see, for example, [Jia+19]), or by random search over the configuration space (as suggested by [BB12]). In the next section, I will demonstrate how this technique can be extended to incorporate a data x meta-parameter interaction, which has yet to be proposed in the literature before.

Interactions between Meta-Parameters and Data Properties

The variance decomposition of the extended grid (presented in Table 3.6) indicates that the meta-parameter learning rate has a minor but noticeable impact on the performance of the classifier obtained from the algorithm. Thus a practitioner implementing this algorithm should invest his search budget in algorithm runs that vary along this parameter. The techniques presented in the following paragraphs can help improve the above guidance by specifying promising meta-parameter value ranges given some algorithmically determinable description or characterization of the data at hand.

I will exemplify this technique by augmenting the extended grid analysis of the previous section with an investigation that should answer whether source sentence length moderates the observed effect of the meta-parameter learning rate. To this end, source sentence length is divided into three bins of $1 - 14$, $15 - 55$, and > 55 words (see section 2.3 for details). In the first step, we modify the LMEM used so far to decompose the variance of a metric by introducing *nested random effects*:

Table 3.8: Variance components with nested random effect in *extended meta-parameter grid* search for neural machine translation model fine-tuned on human marking data [KBR20].

Component	**Variance**	**Percent**
sentence s	0.0562	86.7
residual	0.0074	11.3
learning_rate/sentence_length	0.0012	**1.9**

$$TER = \beta_0 + b_s + \underbrace{b_{learning\_rate} + b_{learning\_rate,sentence\_length}}_{\text{nested random effect:} \quad learning\_rate/sentence\_length} + \epsilon \tag{3.15}$$

where

$$\begin{bmatrix} b_s \\ b_{learning\_rate} \\ b_{learning\_rate,sentence\_length} \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_s^2 & 0 & 0 \\ 0 & \sigma_{l_r}^2 & 0 \\ 0 & 0 & \sigma_{l\_r,s\_l}^2 \end{bmatrix}),$$

$$\epsilon \sim \mathcal{N}(0, \sigma_{residual}^2).$$

And let $\sigma^2_{learning\_rate/sentence\_length} := \sigma^2_{l_r} + \sigma^2_{l\_r,s\_l}$ define the variance of the nested random effect.

Let us take a moment reflecting the model. The only component we haven't encountered so far is $b_{learning\_rate,sentence\_length}$, which denotes a random intercept for every possible combination of learning rate and sentences. Thus the model augments the previous model with a specific learning rate effect for every source sentence length, similar to interactions of fixed effects. The variance decomposition results based on this model are displayed in Table 3.8.

We see that the variance component for learning_rate/sentence_length accounts for 1.9% of the total variance, which is approximately ten times the proportion accounted to learning_rate alone (see Table 3.6). [19] This surge signals a strong interaction between

[19] A careful comparison of both tables shows that the sentence variance sources the additional variance of the nested component caused by sentence length, which introduces a grouping structure among sentences. Hence we start to decompose this component. Nevertheless, this variance transfer only

this meta-parameter and the data property input length.

The former analysis step has given us a macroscopic clue that the length of the input sentences somehow moderates the effect of the learning rate on the algorithm performance. To reveal the microscopic structure of this interaction, we need to rewrite some random effects into fixed effects according to the following model[20] where L is the value set of learning rates used during grid search and S denotes the set of the three length classes:

$$\text{TER} = \underbrace{\text{Learning\_Rate} + \text{Sentence\_Length} + \overbrace{\text{Learning\_Rate x Sentence\_Length}}^{\text{interaction term}}}_{\text{fixed effects}} \tag{3.16}$$

$$+ \underbrace{b^s}_{\text{random intercept}} + \underbrace{\epsilon}_{\text{error}}$$

where

$$\begin{aligned}
\text{Learning\_Rate} &:= \sum_{l \in L} \beta_l \cdot \mathbb{I}_{\{l\}}\left(\text{learning\_rate}\right),\\
\text{Sentence\_Length} &:= \sum_{s \in S} \beta_s \cdot \mathbb{I}_{\{s\}}\left(\text{sentence\_length}\right),\\
\text{Learning\_Rate x Sentence\_Length} &:= \sum_{l \in L} \sum_{s \in S} \beta_{l,s} \cdot \mathbb{I}_{\{s\}}\left(\text{sentence\_length}\right)\\
&\qquad \cdot \mathbb{I}_{\{l\}}\left(\text{learning\_rate}\right)
\end{aligned}$$

$$b_s \sim \mathcal{N}(0, \sigma_s^2) \quad \text{and} \quad \epsilon \sim \mathcal{N}(0, \sigma_{residual}^2).$$

This model is structurally similar to model ((2.16) in section 2.3); therefore, we can apply the same statistical inference techniques as described in section (2.2). Given our interest in studying the microscopic structure of the interaction between learning rate

happens if substantial interaction with the meta-parameter exists. Further, it is also worthwhile to realize that we have left the modeling paradigm given by the initial tautological decomposition (3.3). Consequently, the component associated with sentences should not be interpreted as a reliability coefficient.

[20] Note that this model doesn't contain an intercept. In general, fitting a regression model without an intercept is not advisable. But in the case of this model, which is only based on categorical regressors, it is feasible to fit without an intercept. It is also possible to rewrite it into a form with an intercept. But the notation of this form is more cumbersome. Moreover, inference based on these models yields identical results. Thus I can present the model in this more convenient form.

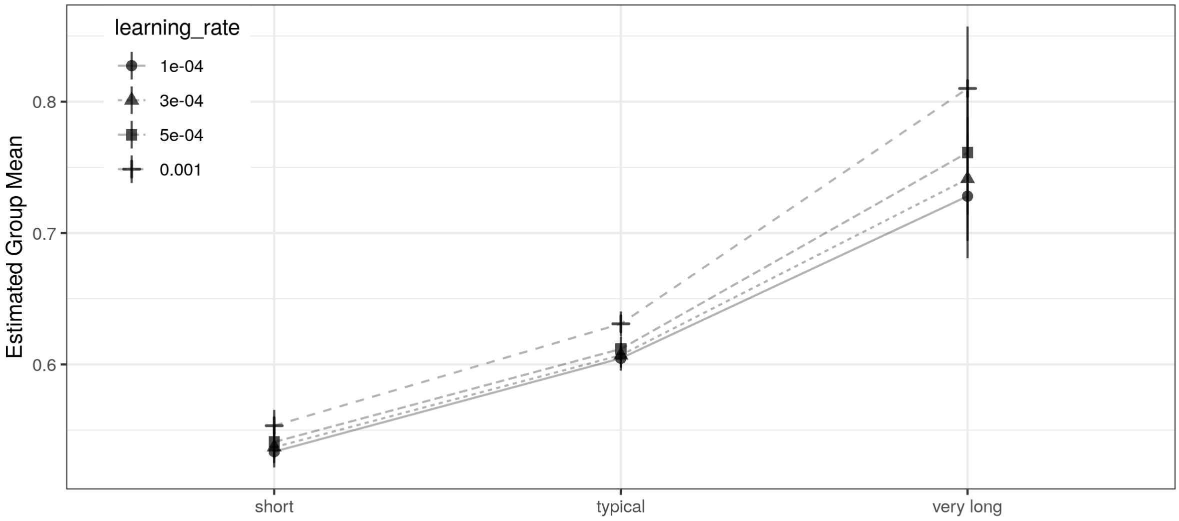

Figure 3.4: Estimated expected translation edit rate (TER) for neural machine translation models trained for different learning rates and input sentence length categories.

and sentence length, we must focus on statistical inference for the model's Learning_Rate x Sentence_Length interaction term. At first, we have to conduct an omnibus test for Learning_Rate x Sentence_Length to test, which yielded a p-value of less than 0.0001. Thus the observed empirical signal is strong enough to justify a present interaction as displayed by the interaction plot in Figure (3.4). A Bonferroni-Holm corrected posthoc test comparing all pairs of learning rates within a length class yields p-values less than 0.0001 for all of them. In general, smaller learning rates show better results than larger ones for all possible inputs, and this gain increases with source sentence length. Thus a machine learning practitioner applying Kreutzer et al. marking method to boost her translation systems performance should spend his search budget on small learning rates, especially when her use case involves longer sentences.

Discussion. The basic components of model-based algorithm performance analysis, including reliability and meta-parameter influence quantification discussed in this chapter, date back to Fisher's [Fis25] statistical techniques for variance component analysis and intra-class correlation coefficients. I replace ANOVA methods with modern LMEMs for modeling and estimation [Woo17] and use refined analytic techniques from psychometrics [Bre01] to show how annotation procedures can be optimized concerning reliability and implementation cost. The psychometric literature includes further reliability measures, which are too plentiful to be covered here. Standard correlation-based reliability coefficients like split-half reliability or the Spearman-Brown formula (see [LN68]) or Cronbach's coefficient alpha [Cro51] can be reformulated as versions of ICCs (see [WSH06]), albeit under variance-restricting conditions that do not seem applicable to either annotators or algorithms. Widely known notions such as inter- or intra-rater reliability can be easily expressed in an ICC-like form (see [Bre01]). For example, in fully crossed designs including facets for raters r, instantiations i, and for interactions sr, si, and ri, with objects of measurement being sentences s, intra-rater reliability is calculated by fixing the rater facet to one rater and generalizing over instantiations, without averaging:

$$\varphi_{intra-rater} = \frac{\sigma_s^2 + \sigma_{sr}^2}{\sigma_s^2 + \sigma_{sr}^2 + \sigma_{ri}^2 + \sigma_{residual}^2}. \tag{3.17}$$

Similar inter-rater reliability is obtained by fixing the instantiations to one and generalizing over raters without averaging:

$$\varphi_{inter-rater} = \frac{\sigma_s^2 + \sigma_{si}^2}{\sigma_s^2 + \sigma_{si}^2 + \sigma_{sr}^2 + \sigma_{residual}^2}. \tag{3.18}$$

However, these measures are formulated in terms of a relative reliability coefficient which is informative when the measurement is used to rank subjects. In contrast, the absolute reliability coefficients are informative when a measurement's value is essential. Thus I have based the definition of φ on this concept.

I have to admit that for the use-case of data annotation, there exists work based on Bayesian data modeling [Pau+18; PC14] that allows a far more elaborated analysis of factors like rater accuracy and behavior or sentence difficulty affecting the annotation process.

Also, variance analytic techniques, often in an ANOVA-like form, have been applied to information retrieval models [FS16; RK12; VSS17] and machine learning models in general [HHL14; ZLH20; BB12][21]. The former approaches focus on interactions between search queries modeled as random effects and retrieval system components modeled as fixed effects. The latter approaches focus on meta-parameter importance without considering interactions between meta-parameter settings and test data properties. But none of the mentioned approaches take advantage of the flexibility of LMEMs to model meta-parameter variance by random effects.

[21] The only exception is [BB12] who apply Gaussian process regression.

A distinctive feature of the proposed approach is the ICC-based idea of quantifying reliability by the proportion of variance attributable to the objects of interest. This simple reliability concept widely applies to algorithm performance analysis. It only requires that the evaluation metric be computed per object (input) and that the test data exhibit sufficient heterogeneity.

In NLP, sentence-level evaluation metrics are often preferred for the interpretability and the ability to calculate sentence-level correlations with human judgments (see [Zha+20; Rei+20] for recent examples). Well, the second condition is nearly always met in all applications. For example, in NLP, high performance on heterogeneous test data is a common requirement to assess the generalization ability of machine learning models in machine translation [Bar+20].

Chapter 4

Inferential Reproducibility

A typical research project in machine learning starts with optimizing a model on given training data, tuning meta-parameters on development data, and ends with evaluating the model using a standard automatic evaluation metric on benchmark test data. For example, a neural machine translation project could use the train-dev-test split of a benchmark parallel dataset provided at `paperswithcode.com` and claim a new SOTA result if a performance difference in BLEU score of a commonly accepted magnitude is achieved over the previous best score published on the leaderboard. By adding the program code (and, if necessary, new data) and publicly sharing meta-parameter settings (following reproducibility checklists[1]), the game can be re-opened to new competitors. This train-dev-test paradigm has greatly fostered research progress in many areas, and allows the researcher to happily focus on improving model performance.

Unfortunately, the party is spoiled by the inherent non-determinism of deep learning that lurks behind randomness in weight initialization, dropout, data shuffling and batching in non-convex optimization [Dau+14; DAm+20], non-determinism due to variations in model architecture and meta-parameter settings [Luc+18; Hen+18], non-determinism due to pre-processing variants and data splits [GB19; Søg+21], and non-determinism due to differences in available computational budget [SGM19; Dod+19]. The problem caused by non-determinism is that slight changes in training settings can reverse relations between baseline and SOTA [RG17; MDB18].

A large body of work identifying similar problems of reproducibility in various machine learning areas has led to claims of a "reproducibility crisis" in AI [Hut18], reverbating a similar crisis in medical sciences [Ioa05]. On the one hand, this has led to considerable efforts to foster reproducibility of SOTA benchmark results in various areas of machine

[1] See, for example, the checklists used at recent issues of NeurIPS (`https://neurips.cc/public/guides/PaperChecklist`) or ACL (`https://aclrollingreview.org/responsibleNLPresearch/`).

learning by raised standards of sharing data, code, and meta-parameter settings [Hei+21; Pin+21; Luc+22]. On the other hand, [Dru09] very early posed the question whether *training reproducibility*[2]— the duplication of a SOTA training result without any changes — is actually interesting and worth having.

In this paper, we propose to embrace the inherent variability of deep learning, and analyze what kind of inferences can be drawn about the strengths and weaknesses of a SOTA model on data with varying characteristics, from models trained under different meta-parameter settings, with the ultimate goal of an application of that model to new data. Such a study falls under the umbrella of *inferential reproducibility.*[3] [GFI16] define it to refer to the drawing of qualitatively similar conclusions from either an independent replication of a study or a reanalysis of the original study. For the case of machine learning, this corresponds to a comparative evaluation of machine learning algorithms themselves, by asking whether a SOTA model yields improvements over a baseline across different settings of meta-parameters and across different characteristics of input data. Furthermore, we want to draw inferences about the sources of variation in model performance, and about their interaction with data characteristics.

The main contribution of our paper is to show how to apply well-known statistical methods to analyze inferential reproducibility. These methods are based on *linear mixed effects models (LMEMs)* fitted to performance evaluation scores of machine learning algorithms. First, we conduct a *generalized likelihood ratio test (GLRT)* to assess statistical significance of performance differences between algorithms, while simultaneously acknowledging for randomness in meta-parameters and data. A key feature is the possibility to assess performance differences conditional on data properties. Second, we show how to use *variance component analysis (VCA)* to facilitate a nuanced quantitative assessment of the sources of variation in performance estimates. Lastly, we compute *reliability coefficient* to assess the general robustness of the model by the ratio of substantial variance out of total variance.

[2] The term was coined by [LÖ22] and corresponds to [Dru09]'s replicability.

[3] This term corresponds to [Dru09]'s reproducibility and was coined by [GFI16]. In order to avoid to contribute further to the terminological confusion in this area (see [Ple18]), we stick to the terms training reproducibility and inferential reproducibility in the following.

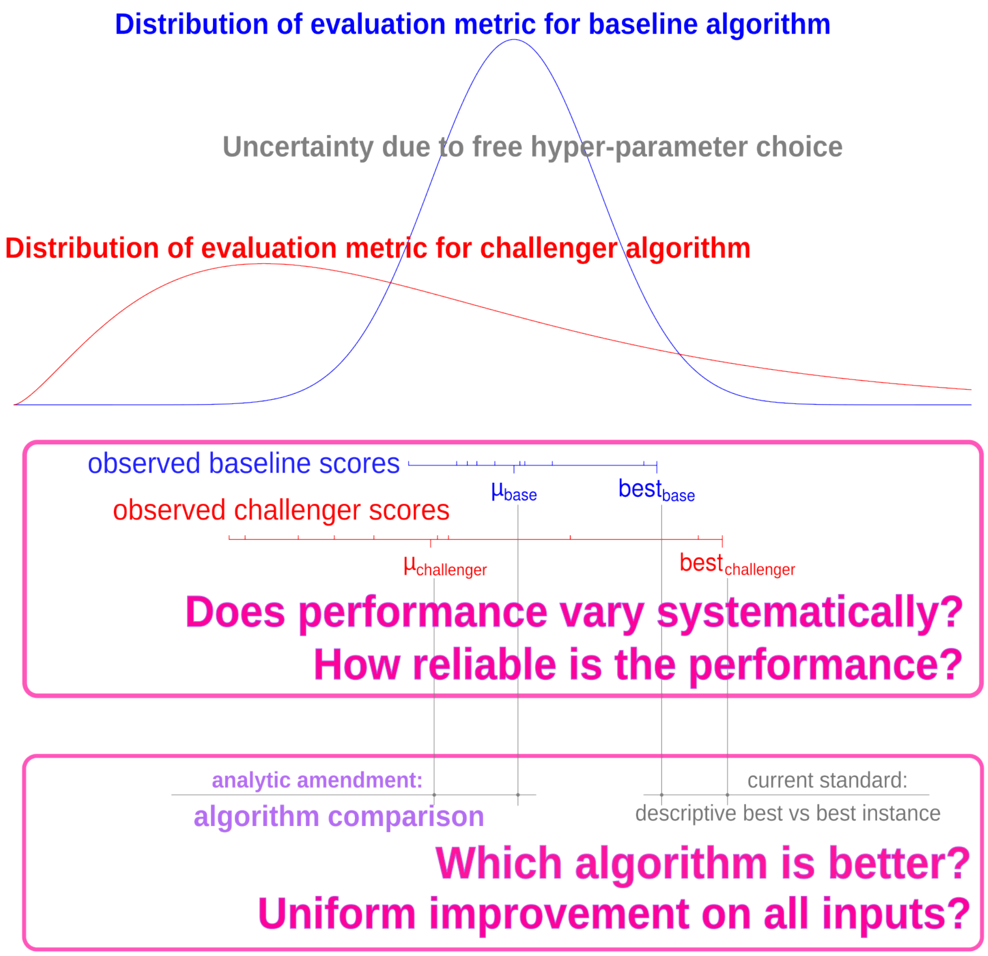
Distribution of evaluation metric for baseline algorithm
Uncertainty due to free hyper-parameter choice
Distribution of evaluation metric for challenger algorithm
observed baseline scores
μ_{base}
$best_{base}$
observed challenger scores
$\mu_{challenger}$
$best_{challenger}$
Does performance vary systematically?
How reliable is the performance?
analytic amendment:
algorithm comparison
current standard:
descriptive best vs best instance
Which algorithm is better?
Uniform improvement on all inputs?

Figure 4.1: Performance comparison between baseline (blue curve) and SOTA (red curve) with respect to training reproducibility (best baseline versus best SOTA result) and inferential reproducibility (comparison of estimated expected performance under meta-parameter variation). Here relations are reversed if meta-parameter variation is taken into account, requiring a closed look with the magnifying glass of a reliability analysis.

4.1 A Scheme for Analyzing Inferential Reproducibility

Figure 4.1 shows the distributions of performance evaluation scores of a baseline system (blue curve) and a SOTA system (red curve). The depicted scenario is one where the best evaluation result for the SOTA model is located in the long tail of the score distribution and thus only found by extensive meta-parameter search. We define the assessment of statistical significance of the difference of the best SOTA results against that of the best score of the baseline system as the task of training reproducibility. However, such a test fails to assess the systematic aspect of comparing the machine learning algorithms themselves. We define this to be the task of inferential reproducibility. This can be achieved by a significance test on the system effect parameter of an LMEM that estimates the means of the distributions of performance scores obtained under meta-parameter variation of SOTA and baseline model, and can be conditioned on data properties. In the depicted scenario, the relations between SOTA and baseline are reversed if meta-parameter variation is taken into account. LMEMs allow us to conduct a deeper analysis to quantify the sources of randomness and variability in performance evaluation. This is done by a reliability analysis (shown by the magnifying glass) that analyzes the variance contributed by meta-parameters and their interaction with data properties.

4.2 Linear Mixed Effects Models

A linear mixed effects model (LMEM) is an extension of a standard linear model that allows a rich linear structure in the random component of the model, where effects other than those that can be observed exhaustively (so-called *fixed effects*) are treated as a random samples from a larger population of normally distributed random variables (so-called *random effects*).

Given a dataset of N input-output pairs $\{(\mathbf{x}^n, y^n)\}_{n=1}^N$, the general form of an LMEM is

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\epsilon}, \tag{4.1}$$

where $\mathbf{X}$ is an $(N \times k)$-matrix and $\mathbf{Z}$ is an $(N \times m)$-matrix, called model- or design-matrices (both are known), which relate the unobserved vectors $\boldsymbol{\beta}$ and $\mathbf{b}$ to $\mathbf{Y}$. $\boldsymbol{\beta}$ is a k-vector of fixed effects and $\mathbf{b}$ is an m-dimensional random vector called the random effects vector. $\boldsymbol{\epsilon}$ is an N-dimensional vector called the error component. The random vectors are assumed to have the following distributions:

$$\mathbf{b} \sim \mathcal{N}(0, \psi_\theta), \; \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Lambda}_\theta), \tag{4.2}$$

where ψ_θ and $\boldsymbol{\Lambda}_\theta$ are covariance matrices parameterized by the vector θ.

The most common application of LMEMs is to model complex covariance structures in the data when the usual i.i.d. assumptions fail to be applicable. This is the case for repeated or grouped, and thus non-independent, measurements such as multiple ratings of same items and same subjects in psycho-linguistic experiments. LMEMs have become popular in this area due to their flexibility [BDB08; Bat+15], and have even been credited as candidates to replace ANOVA [Bar+13]. The price for this flexibility is an elaborate estimation methodology for which we refer the reader to further literature [PB00; MS01; WWG07; Dem13; Woo17].

4.3 Generalized Likelihood Ratio Tests w/ and w/o Measurement Variation

Let us assume our goal is to test the statistical significance of an observed performance difference between a baseline and a SOTA system. For the sake of concreteness, let us assume we are comparing Natural Language Processing (NLP) models on a benchmark test set of gold standard sentences. In order to conduct a generalized likelihood ratio test (GLRT) for this purpose, we need to fit two LMEMs on the performance evaluation data of baseline and SOTA system which analyze the data differently, and compare their likelihood ratio. Let us further assume an experimental design where variants of the baseline and SOTA models, corresponding to different meta-parameter configurations during training, are evaluated on the benchmark data. Simple linear models are a suboptimal choice to analyze this experiment since they are based on the assumption that each system was evaluated once on a disjoint set of sentences. This would force us to average over variants, thereby losing useful information contained in the clusters of repeated measurements of the same test input.

LMEMs allow us to better reflect this design and to leverage its statistical benefits by adding a random effect b_s for each sentence in our evaluation model. Such a model decomposes the total variance of the evaluation score into three blocks: systematic variance due to the fixed effects of the model, variance due to sentence heterogeneity, and unexplained residual variance. This allows us to reduce the as of yet unaccounted residual variance by attributing a variance component σ_s^2 to variance between sentences. If we think of the residual error as noise that masks the signal of measured performance scores, we can effectively perform a noise reduction that increases the power of our tests to detect significant differences.

A straightforward technique to implement statistical significance tests using LMEMs is the so-called *nested models* setup [PB00]. First we train an LMEM that doesn't distinguish between systems. This restricted model

$$m_0 : Y = \beta + b_s + \epsilon_{res} \tag{4.3}$$

specifies a common mean β for both systems as fixed effect, and a sentence-specific deviation b_s as random effect with variance σ_s^2, and a residual error ϵ_{res} with variance σ_{res}^2 for the performance scores Y. It represents the null hypothesis that there is no difference between systems. This model is compared to a more general model that allows different means for baseline and SOTA scores:

$$m_1 : Y = \beta + \beta_c \cdot \mathbb{I}_c + b_s + \epsilon_{res} \tag{4.4}$$

This model includes an indicator function $\mathbb{I}_c$ to activate a fixed effect β_c that represents the deviation of the competing SOTA model from the baseline mean β when the data point was obtained by a SOTA evaluation. The restricted model is a special case of this model (thus "nested" within the more general model) since it can be obtained by setting β_c to zero. Let ℓ_0 be the likelihood of the restricted model, and ℓ_1 be the likelihood of the more general model, the intuition of the likelihood ratio test is to reject the null hypothesis of no difference between systems if the statistic

$$\lambda = \frac{\ell_o}{\ell_1} \tag{4.5}$$

yields values close to zero.

The incorporation of a random sentence effect b_s introduces a pairing of systems on the sentence level that corresponds to standard pairwise significance tests. However, clustering at the sentence level allows accounting for arbitrary kinds of uncertainty introduced by the random nature of the training process. This setup is thus not only suitable for pairwise comparisons of best baseline and best SOTA model in order to test training reproducibility, but it also allows incorporating broader variations induced by meta-parameter settings of baseline and SOTA systems, thus making it suitable to test inferential reproducibility.

A further distinctive advantage of GLRTs based on LMEMs is that this framework allows analyzing significance of system differences conditional on data properties. For example, we could extend models m_0 and m_1 by a fixed effect β_d modeling a test data property d like readability of an NLP input sequence, or rarity of the words in an input sequence, and by an interaction effect β_{cd} allowing to assess the expected system performance for different levels of d. The enhanced model

$$m_1' : Y = \beta + \beta_d \cdot d + (\beta_c + \beta_{cd} \cdot d) \cdot \mathbb{I}_c + b_s + \epsilon_{res} \tag{4.6}$$

would then be compared to a null hypothesis model of the form

$$m_0' : Y = \beta + \beta_d \cdot d + b_s + \epsilon_{res}. \tag{4.7}$$

GLRTs belong to the oldest techniques in statistics, dating back to [NP33; Wil38]. For more information on extensions of GLRTs for multiple comparisons and on their asymptotic statistics we refer the reader to further literature [Vaa98; PB00; Paw01; Dav03; LM12].

4.4 Variance Component Analysis and Reliability Coefficients

The main goal of a reliability analysis in the context of a reproducibility study is to quantify and analyze the sources of randomness and variability in performance evaluation, and to quantify the robustness of a model in a way that allows to draw conclusions beyond the concrete experiment. The first goal can be achieved by performing a variance component analysis (VCA). For example, let us assume we want to specify a model for performance evaluation scores that besides a global mean μ specifies random effects to account for variations in the outcome Y specific to different sentences s and specific to different settings of a regularization parameter r. A tautological decomposition of the response variable into the following four components can be motivated by classical ANOVA theory [SCM92; Bre01]:

$$Y = \mu + (\mu_s - \mu) + (\mu_r - \mu) + (Y - \mu_s - \mu_r + \mu). \tag{4.8}$$

The components of the observed score Y for a particular regularization setting r on a single sentence s are the grand mean μ of the observed evaluation score across all levels of regularization and sentences; the deviation $\nu_s = (\mu_s - \mu)$ of the mean score μ_s for a sentence s from the grand mean μ; the deviation $\nu_r = (\mu_r - \mu)$ of the mean score μ_r for a regularization setting r from the grand mean μ; and the residual error, reflecting the deviation of the observed score Y from what would be expected given the first three terms. Except for μ, each of the components of the observed score varies from one sentence to another, from one regularization setting to another, and from one regularization-sentence combination to another. Since these components are uncorrelated with each other, the total variance $\sigma^2(Y-\mu)$ can be decomposed into the following *variance components*:

$$\sigma^2(Y-\mu) = \sigma_s^2 + \sigma_r^2 + \sigma_{res}^2, \tag{4.9}$$

where σ_s^2 and σ_r^2 denote the variance due to sentences and regularization settings, and σ_{res}^2 denotes the residual variance component including the variance due to interaction of s and r.

Let $\nu_f = \mu_f - \mu$ denote a deviation from the mean for a facet[4] f whose contribution to variance we are interested in. Instead of estimating the corresponding variance components σ_f by ANOVA expected mean square equations, we use LMEMs to model each ν_f as a component of the random effects vector $\mathbf{b}$ in (4.2), and model each corresponding variance component σ_f^2 as an entry of the diagonal variance-covariance matrix ψ_θ in (4.2). Besides greater flexibility in estimation[5], LMEMs also allow analyzing the interaction of meta-parameters and data properties. This can be achieved, for example, by changing the random effect b_r to a fixed effect β_r, and by adding a fixed effect β_d modeling test data characteristics, and an interaction effect β_{rd} modeling the interaction between data property d and choice of meta-parameter r.

The final ingredient of a reliability analysis is the definition of a coefficient that relates variance components to each other, instead of inspecting them in isolation. The key concept is the so-called *intra-class correlation coefficient (ICC)*, dating back to [Fis25]. A fundamental interpretation of the ICC is as a measure of the proportion of variance that is attributable to substantial variance, i.e., to variance between the objects of measurement. The name of the coefficient is derived from the goal of measuring how strongly objects in the same class are grouped together in a measurement. Following [Bre01], we can define a concrete reliability coefficient, denoted by φ, for our application scenario. In our case, objects of interest are test sentences s, and substantial variance is variance σ_s^2 between sentences. Assume facets $f_1, f_2, \ldots$ and selected interactions $sf_1, sf_2, f_1f_2, \ldots$. Then the

[4] In the psychometric approach to reliability of [Bre01], the conditions of measurement that contribute to variance in the measurement besides the objects of interest are called *facets* of measurement. In our running NLP example, the objects of interest in our measurement procedure are the sentences. They are the essential conditions of measurement. The only facet of measurement in this example are the regularization settings, while the objects of interest are not usually called a facet.

[5] Among the many advantages of using LMEMs to estimate variance components is that the same model structure can be used for designs that are special cases of the fully crossed design, and the elegant handling of missing data situations. See [BDB08; Bar+13; Bat+15] for further discussions on the advantages of LMEMs over mixed-model ANOVA estimators.

reliability coefficient φ is computed by the ratio of substantial variance σ_s^2 to the total variance, i.e., to itself and the error variance σ_Δ^2 that includes variance components for all random effects and selected interactions of random effects:

$$\varphi = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_\Delta^2}, \text{ where } \sigma_\Delta^2 = \sigma_{f_1}^2 + \sigma_{f_2}^2 + \ldots + \sigma_{sf_1}^2 + \sigma_{sf_2}^2 + \ldots + \sigma_{f_1 f_2}^2 + \cdots + \sigma_{res}^2. \tag{4.10}$$

Based on this definition, reliability of a performance evaluation across replicated measurements is assessed as the ratio by which the amount of substantial variance outweighs the total error variance. That is, a performance evaluation is deemed reliable if most of the variance is explained by variance between sentences and not by variance within a sentence, such as variance caused by random regularization settings or by residual variance due to unspecified facets of the measurement procedure. Naturally, different assumptions on thresholds on this ratio will lead to different assessments of reliability. A threshold of 80% is used, for example, by [Jia18]. Values less than 50%, between 50% and 75%, between 75% and 90%, and above 90%, are indicative of poor, moderate, good, and excellent reliability, respectively, according to [KL16].

VCA and ICCs date back to the works of [Fis25]. More information can be found in [SF79; SCM92; MW96; Bre01; WSH06].

4.5 A Worked-Through Example

We exemplify the introduced methodes by an NLP example[6], namely BART+R3F fine-tuning algorithm presented by [Agh+21] for the task of text summarization, evaluated on the CNN/DailyMail [Her+15] and RedditTIFU [Kim+19] datasets.

BART+R3F was listed as SOTA on these datasets at the time of writing. It uses an approximate trust region method to constrain updates on embeddings f and classifier g during fine-tuning in order not to forget the original pre-trained representations. This is done by minimizing a task loss $\mathcal{L}(\theta)$ regularized by the Kullback-Leibler distance on normally or uniformly distributed parameters:

$$\mathcal{L}(\theta) + \lambda KL_{sym}(g \cdot f(x) || g \cdot f(x+z)) \text{ s.t. } z \sim \mathcal{N}(0, \sigma^2 I) \text{ or } z \sim \mathcal{U}(-\sigma, \sigma). \tag{4.11}$$

The first question we want to answer is that of training reproducibility – is the result

[6] This example can be found on `paperswithcode.com` open resource.

Table 4.1: Text summarization results (Rouge-1/2/L) for baseline (BART) and SOTA (BART+R3F) reported in [Agh+21].

	CNN/DailyMail	RedditTIFU
baseline	44.16/21.28/40.90	24.19/8.12/21.31
SOTA	44.38/21.53/41.17	30.31/10.98/24.74

Table 4.2: Significance of result difference baseline-SOTA on CNN/DailyNews.

	baseline	SOTA	p-value	effect size
Rouge-1	44.09	44.41	< 0.0001	-0.101
Rouge-2	21.13	21.44	< 0.0001	-0.080
Rouge-L	40.81	41.16	< 0.0001	-0.105

difference between baseline and new SOTA reproducible on the data[7] and the code[8] linked on the repository, and under the meta-parameter and preprocessing setup reported in the paper. As baseline we take a pre-trained BART-large[9] model [Lew+20]. The Rouge-1/2/L[10] [LH03] results for the text summarization task reported in [Agh+21] are shown in Table 4.1.

Let us first look at the results on the CNN/DailyMail dataset. The paper gives detailed meta-parameter settings for the text summarization experiments, but reports final results as maxima over training runs started from 10 unknown random seeds. Furthermore, the regularization parameter is specified as a choice of $\lambda \in [0.001, 0.01, 0.1]$, and the noise type as a choice from $[\mathcal{U}, \mathcal{N}]$. Using the given settings, we started the BART+R3F code from 5 new random seeds and the BART-large baseline from 18 random seeds on 4 Nvidia Tesla V100 GPUs each with 32 GB RAM and a update frequency of 8. All models were trained for 20-30 epochs using a loss-based stopping criterion. Searching over the given meta-parameter choices, we obtained a training reproducibility result given in Table 4.2: We find significant improvements of the best SOTA model over the best baseline with respect to all Rouge-X metrics (the difference baseline - SOTA is negative). However, the effect sizes (standardized mean difference between evaluation scores) are small.

Let us next inspect significance conditional on data properties. We quantify properties

[7] `https://github.com/abisee/cnn-dailymail`, `https://github.com/ctr4si/MMN`

[8] `https://github.com/facebookresearch/fairseq/tree/main/examples/rxf`

[9] `https://github.com/facebookresearch/fairseq/tree/main/examples/bart`

[10] We used the module `files2rouge` (v2.1.0 downloaded April 2022) with default parameters to calculate Rouge-1/2/L scores. This module provides a wrapper function for the `ROUGE-1.5.5` perl script released by [LH03].

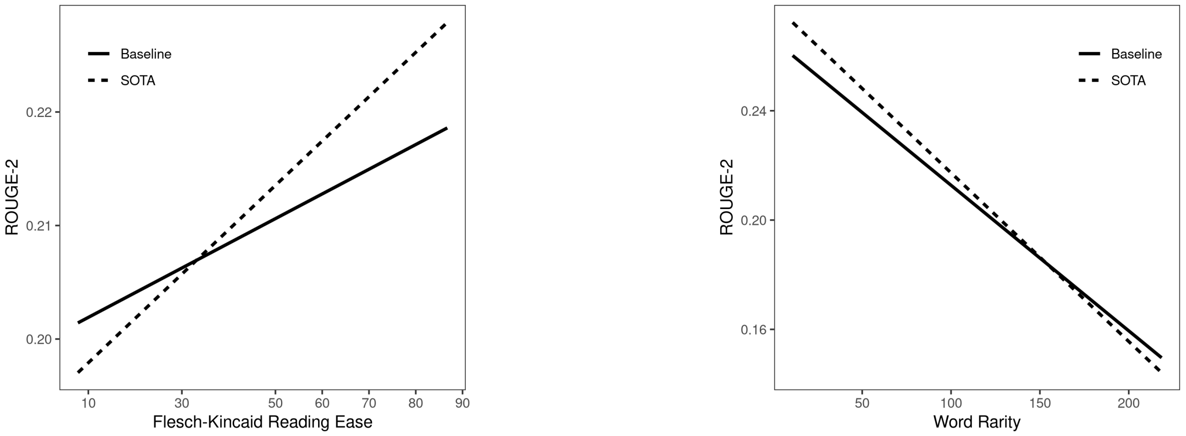

Figure 4.2: Interaction of Rouge-2 of baseline (solid) and SOTA (dashed) with readability (left) and word rarity (right).

of summarization inputs by word rarity [Pla+19], i.e., the negative logarithm of the empirical probabilities of words in summary inputs, where higher values mean higher rarity. Furthermore, we calculate readability [Kin+75] of summary inputs by calculating the ratio of words/sentences and syllables/word. Readability scores are in principle unbounded, however, an interpretion scheme exists for the range from 0 (difficult) to 100 (easy). An analysis of significance conditional on data properties can be seen as first step of inferential reproducibility. The interaction plots given in Figure 4.2 show a significant difference in performance slope for Rouge-2 with respect to ease of readability, where the performance of the best SOTA system increases faster than that of the best baseline for easier inputs (left plot). Also, a significant difference in Rouge-2 with respect to word rarity is seen where the best SOTA model is better than the best baseline for inputs with lower word rarity (right plot).

The next question of inferential reproducibility is whether the results given above are robust against meta-parameter variations, and which meta-parameters are most important in order to achieve the best result. We inspect the original grid of meta-parameter configurations of the SOTA model, given by crossing the given choices of meta-parameters with each other, yielding 3 $\lambda \times 2$ noise distributions $\times$ 5 random seeds $= 30$ configurations. As shown in Table 4.3, the relations between SOTA and baseline are turned around (the difference baseline - SOTA is positive) showing significant wins of baseline over SOTA at medium effect size.

Since the performance variation of the baseline model over 18 random seeds was negligible (standard deviations $< 0.2\%$ for Rouge-X scores), we conduct a reliability analysis of the SOTA model in order to reveal the culprit for this performance loss. The

Table 4.3: Significance of baseline-SOTA on CNN/DailyNews under meta-parameter variation.

	baseline	SOTA	p-value	effect size
Rouge-1	44.15	42.21	< 0.0001	0.390
Rouge-2	21.26	19.64	< 0.0001	0.301
Rouge-L	40.84	38.53	< 0.0001	0.531

Table 4.4: Variance component analysis for Rouge-1 (top), Rouge-2 (middle), and Rouge-L (bottom) estimates.

Variance component v	Variance σ_v^2	Percent
summary_id	0.00923	**55.8**
lambda	0.00254	15.0
random_seed	0.00012	0.7
noise_distribution	0.00005	0.3
residual	0.00464	27.1
summary_id	0.00992	**62.7**
lambda	0.00131	8.3
random_seed	0.00008	0.5
noise_distribution	0.00003	0.2
residual	0.00449	28.3
summary_id	0.00875	**47.9**
lambda	0.00519	28.4
random_seed	0.00004	0.2
noise_distribution	0.00001	0.1
residual	0.00428	23.4

variance component analysis in Table 4.4 shows that the variance contributions due to variation in random seeds or choice of noise distribution are negligible. However, in all three cases the largest contribution to variance is due to the regularization parameter λ. The percentage of variance due to objects of interest, here summaries, can readily be interpreted as reliability coefficient φ, yielding moderate reliability for performance evaluation under Rouge-1 and Rouge-2 (φ between 50% and 75%) and poor reliability for evaluation under Rouge-L (φ below 50%).

An inspection of the interaction of data properties with the regularization parameter is given in Figure 4.3. The interaction plots show a significant drop in Rouge-2 performance of the SOTA model for the regularization parameter $\lambda = 0.1$ across all levels of reading ease (top plot) and for rare words (bottom plot).

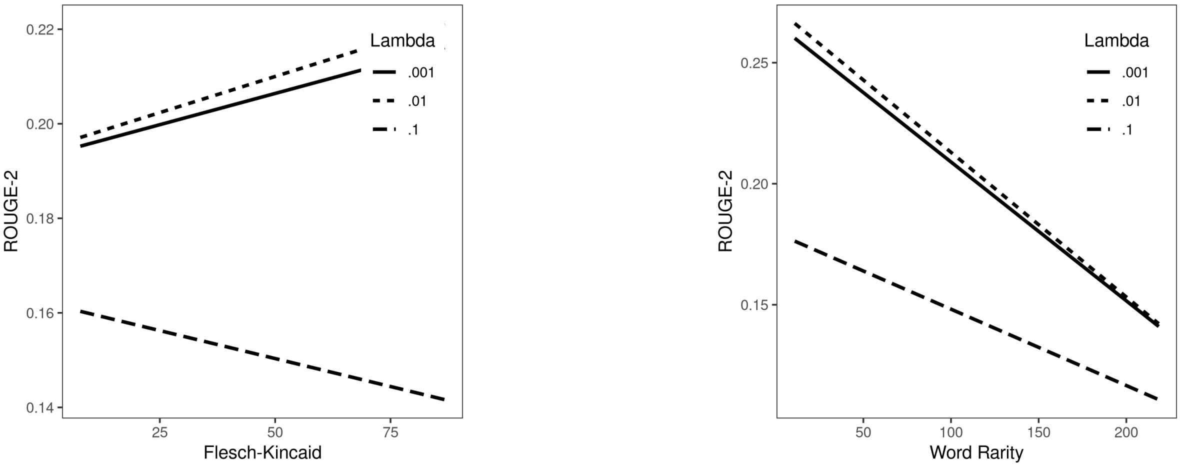

Figure 4.3: Interaction of Rouge-2 of SOTA for different values of regularization parameter λ with readability (left) and word rarity (right).

Let us inspect the results on the RedditTIFU dataset next. These data are interesting since they are much harder to read (mean readability score of -348.9), however, a reproducibility analysis on the RedditTIFU dataset was hampered by the fact that the train/dev/test split for RedditTIFU data (long version) was not given on `paperswithcode.com` nor reported in the paper or the code. We used the split[11] provided by [Zho+20] instead. Under this data split, we found a significant improvement of the best SOTA over the best baseline at a small effect size (-0.155) only for Rouge-2. If meta-parameter variation was taken into account, the effect size was even smaller (-0.0617). There were no significant interaction effects and neglible variance contributions from meta-parameters.

In sum, this small study allows a nuanced assessment of the strengths and weaknesses of the BART+R3F model: Losing or winning a new SOTA score strongly depends on finding the sweet spot of one meta-parameter (here: λ), while the paper's goal was explicitly to reduce instability across meta-parameter settings. Performance improvements by fine-tuning are achieved mostly on easy-to-read and frequent-word inputs – these comprise less than one quarter of the CNN/Dailynews data. Lastly, the model does not seem to be robust against variations in data – under a new random split on RedditTIFU the large gains reported for the split used in the paper can no longer be achieved.

[11] `https://paperswithcode.com/sota/text-summarization-on-reddit-tifu`

4.6 Related Work

Discussions of reproducibility problems in research date back at least to [Ioa05], and for the area of machine learning at least to [Han06]. Since then, a multitude of papers has been published on new aspects of the problem and new exemplifications, however, much less work has been invested in concrete techniques to solve the problem.

For example, special-purpose significance tests have been proposed for particular evaluation metrics ([Dro+20], Chapter 3), for meta-parameter variations [DSR19], and for multiple test data [Dro+17]. One advantage of the proposed LMEM-based approach is that it unifies these special-purpose techniques into a single framework for hypothesis testing. Furthermore, extensions of bootstrap [Sel+21; Bou+21] or permutation [Cla+11] tests have been proposed to incorporate meta-parameter variation. The distinctive advantage of our approach is that it enables analyzing significance of result differences conditional on data properties. These can be generic data properties like readability as above, or properties of combined datasets obtained from different sources like data splits, bootstrapped data, or different-domain data sets.

Variance component analysis based on ANOVA techniques has been applied to information retrieval models [FS16; VSS17] and machine learning models in general [HHL14; ZLH20]. These approaches focus on meta-parameter importance and ignore an incorporation of data variability into their analysis. We replace ANOVA methods by LMEMs for modeling and estimation [Woo17] and promote the ICC-based idea of quantifying reliability by the proportion of variance attributable to the objects of interest, which to our knowledge has not been applied to machine learning before.

From a broader perspective, our work can be seen as a contribution to trustworthiness [Hua+20] and interpretability [Zha+21] of deep neural networks, with a focus on understanding variability in the performance of a neural network depending on data characteristics, meta-parameter settings, and their interactions.

4.7 Discussion

Widely recognized work by statisticians has proposed to abandon statistical significance testing, at least its role in screening of thresholds and as guarantor of reproducibility, but instead to report continuous p-values, along with other factors such as prior evidence [GL14; Col17; McS+19]. Our proposed use of GLRTs, VCA and ICCs aligns with these recommendations. Our focus is to use them as analysis tools of model performance under different meta-parameter settings, dependent on characteristics of data, and to detect the sources of variance and their interactions with data properties. This allows us to address questions of genuine interest to researchers and users like "Will the SOTA algorithm's stellar performance on the benchmark testdata lead to top performance the kinds of datasets that my customers will bring?", or more specifically "How will individual test example characteristics or particular meta-parameter settings, and their interaction with data properties, affect performance?" Our methods are readily applicable to performance evaluation data already obtained during meta-parameter optimization. They allow us to transform this usually unused data into new findings about algorithm behavior. We believe that they will be especially useful for large-scale experiments where a manual inspection of variance due to interactions of large numbers of meta-parameters and data properties is prohibitive.

Chapter 5

Concluding Remarks

The statistical methods presented in this work greatly enhance the scope of conclusions drawn from a machine learning experiment. Instead of discarding most of the information created during an investigation for the evaluation, as done for most research following the current de facto standard, the proposed techniques can leverage all the data produced for a more elaborated and empirically grounded algorithm analysis.

The basic approach to use established statistical methods successfully applied in a wide range of empirical sciences and adapt them to facilitate a more principled analysis of machine learning experiments is not new [Die98; Hot+05; Dro+17; DSR19; Dro+20; Bou+21]. A common thread of all these attempts is the adaptation of particular significance tests for group comparisons. A principle limitation of this approach is that these tests can only incorporate one source of randomness (typically) due to data sampling. Applied to contemporary deep learning experiments where the analyst is confronted with several sources of randomness, e.g., sampled training, development and test data, meta-parameters sampled from the meta-parameter space of the algorithm, the deliberate introduction of randomness in the learning (optimization) process to facilitate and improve it, these tests are bound to capture only one aspect of the actual variability. Thus they allow only the analysis of very constrained experiments, thereby implicitly limiting the scope of conclusion that can be drawn. A fact that is often ignored and never made explicit.

Some authors [Bou+21] seem to be aware that limited experimental designs[1] allow only limited conclusions and advocates rightly that it is necessary to vary and include as many aspects as possible in the analysis to get a comprehensive picture of an algorithm's performance. However, they failed to recognize that such a comprehensive design introduces

[1] To avoid confusion here, one should be aware that this term does not indicate the experimenter's actions. It characterizes only the principle structure of the data that enters the analysis.

a non-negligible none-iid stochastic structure in the evaluation data. The iid violation is a consequential problem because nearly all classical test construction principles, whether parametric, non-parametric, or sampling-based, and all theoretical guarantees assume it.

From a technical point of view, the methods suggested in this work allow the construction of analytical instruments for non-iid sample structures, thus expanding previous research. The primary technological devices for experimental analysis presented in this work are Linear Mixed Effects models (LMEM) [Woo17] and the Generalized Likelihood Ratio Test (GLRT) [NP33; Wil38]. LMEMs are one of the most flexible classes of regression models. Like Generalized Linear Models [MN89], they allow to build models for random variables of the exponential family, but due to their ability to include random effects also provide a flexible interface to model complex non-iid data structures. The estimation and inference theory for LMEM is [PB00; Woo03; Woo17; GO20] is well developed and implemented in popular programming languages [BMB12; Bat+15]. Together LMEMs and the GLRT provide a unified framework for algorithm analysis and comparison, which I call *model-based* analysis.

Before discussing this approach's merits further, I want to clarify the function that models perform in machine learning and statistics. In machine learning, the role of a model is to provide accurate outputs for previously unseen input instances, whereas, in statistics, models are used to study the relation between variables. For the evaluation of machine learning experiments, models are used in the "statistical" way to learn the connection between the variables "Performance" and "Algorithms" (in the case of algorithm comparison) or "Meta-parameter(s)" (in the case of algorithm performance analysis).

The mechanics of a model-based algorithm comparison is using LMEMs to build hypotheses-specific data models, trained on test data evaluation scores of machine learning systems, and the GLRT for statistical inference. This setup allows the incorporation of variability into significance testing by blocking repeated measurements obtained from different meta-parameter configurations on the input unit level (e.g., source sentences). Thus, it allows accounting for uncertainty introduced by the random nature of the training process. The clustering of repeated measurements on the input unit level is universally applicable. The instances don't need to share the same meta-parameter configuration or space. This principle is applicable even when the number of replications is entirely different for each algorithm. The clustering of repeated measurements on the sentence level is a default option that can be extended if systems share other facets of variation, for example, if they are minor variations of each other and share different meta-parameter facets. Furthermore, similar to analyzing the dependency on test sentence properties, indicators for subsets can be used to analyze the dependence on domains or topics.

The central analytical instrument of a model-based algorithm performance analysis is an LMEM implementation of a variance decomposition of the evaluation data of an algorithm. Adapting concepts well established in psychometrics for decades, I defined the coefficient φ as a measure of algorithm performance reliability. Reporting φ complementary to the best-achieved performance of a machine learning algorithm on a test set gives an impression about the necessary computational budget needed to be invested in the meta-parameter search to obtain it. A low value of φ reveals large computational requirements to find the best-performing instance settings. In contrast, large values of φ indicate an insensitivity of the algorithm concerning meta-parameter choices.

Further, variance component analysis of algorithm performance allows us to assess the importance of meta-parameters and the interaction of meta-parameters and data. For the latter study, one can turn the hyper-parameter/data-property interactions of interactions into fixed effects for a detailed analysis. The benefits of this analysis are the possibility of freezing meta-parameters with a negligible contribution to the total variance, which will aid more efficient meta-parameter optimization that makes the best use of a given computational budget and knowing which hyper-parameters indeed cause substantial performance changes could also shed light on the actual workings of an algorithm.

The chapter on inferential reproducibility combines algorithm performance analysis and comparison in a showcase re-analysis of an experiment conducted by Aghajanyan et al. [Agh+21] to demonstrate the benefits of model-based algorithm analysis. In this experiment, the authors suggested an improved fine-tuning method (termed BART-R3F) that introduces a penalty term in the objective function so that large deviations from the initial model in the weight space received are penalized. The author claims that while this method prevents catastrophic forgetting [Fre99] also benefits performance. Replicating the experiment according to the study protocol and reanalyzing it yields a mixed picture. In summary, we were able to replicate the numbers reported by Aghajanyan et al. for the CNN/Dailynews data set but not for RedditTIFU, where we had to resort to the official RedditTIFU data split because the authors did not publish information about the split they have used. We confirmed that the best models are slightly but practically negligible better for one data. Further analysis revealed that BART-R3F requires an extensive and costly meta-parameter search to show these marginal gains and that these improvements were only noticeable for the subset of easy examples. The latter observation also explains why BART-R3F cannot show progress on the RedditTIFU data set, composed chiefly of difficult examples. Several findings confirmed BART-R3F's sensitivity concerning meta-parameter choice. Firstly, on an algorithmic level, BART-R3F fails to show improvements and, indeed, performs worse than the baseline on average. Secondly, BART-R3F has low values of φ for all evaluation metrics. A detailed analysis of the essential hyper-parameter λ revealed that the performance of BART-R3F is better the smaller λ is, effectively limiting the impact of the penalty term in the objective. This analysis exemplifies how incomplete and misleading a status-quo descriptive best vs. best comparison is. Without much efford, a model-based evaluation can give a richer, more detailed, empirically grounded, and easily communicable summary of an experiment.

An advantage of the presented model-based techniques is that they apply to arbitrary models and tasks[2]. The last remark concerns the principle structure of the evaluation metrics with the presented methods. We must distinguish between metrics expressed as an expectation (or mean) of a loss-like function, e.g., expected TER score, and metrics aggregated in a different we, e.g., corpus BLEU. Generally, the presented methods can be applied to the former rather than the latter.

[2] For tasks with non-numeric outputs, a numeric evaluation metric (e.g., edit distance in our machine translation examples) that scores the accuracy of the predicted output can produce the desired numeric information.

List of Figures

1 Introduction 1

2 Algorithm Comparison 7

2.1 Critical region of two-tailed Z-test. 15

2.2 Critical region of left-tailed Z-test. 16

2.3 p-value based on χ^2 distribution. 38

2.4 Median TER scores for baseline and machine translation systems fine-tuned on markings or post-edits. 41

2.5 TER scores for baseline and machine translation systems fine-tuned on markings or post-edits, plotted against source sentence length. 49

2.6 TER scores for baseline and machine translation systems fine-tuned on markings or post-edits, plotted against three bins of source sentence length. 49

2.7 Interaction plot of estimated TER scores for baseline and machine translation systems fine-tuned on markings or post-edits and bins of source sentence length. 51

3 Algorithm Performance Analysis 55

3.1 Mean and 95% confidence intervals for F1-score, precision, recall of GANs for different computational budgets. Graphics from [Luc+18]. 68

3.2 Sentence-wise quality judgment score computed as the ratio of marked or edited words per sentence, for 3 rating instantiations of 5 sentences by each of 10 raters. 80

3.3 Reliability coefficient φ for data annotation performance in "Marking" or "Post Edit" mode, generalized to 5 rating instantiations and 12 raters. . 82

3.4 Estimated expected translation edit rate (TER) for neural machine translation models trained for different learning rates and input sentence length categories. 91

4 Inferential Reproducibility 95

4.1 Performance comparison between baseline (blue curve) and SOTA (red curve) with respect to training reproducibility (best baseline versus best SOTA result) and inferential reproducibility (comparison of estimated expected performance under meta-parameter variation). Here relations are reversed if meta-parameter variation is taken into account, requiring a closed look with the magnifying glass of a reliability analysis. 97

4.2 Interaction of Rouge-2 of baseline (solid) and SOTA (dashed) with readability (left) and word rarity (right). 105

4.3 Interaction of Rouge-2 of SOTA for different values of regularization parameter λ with readability (left) and word rarity (right). 107

5 Concluding Remarks 111

Bibliography 119

List of Tables

1 Introduction 1

2 Algorithm Comparison 7

2.1 Notation for multiple testing 24

2.2 Effect of evaluation strategy on estimated error variance and significance. 46

2.3 p-values for pairwise TER differences between systems on the test set. . . 48

2.4 p-values for pairwise differences between systems on source sentences of different lengths. 51

3 Algorithm Performance Analysis 55

3.1 Variance components in translation marking experiment. 81

3.2 Variance components in translation post-editing experiment. 81

3.3 Meta-parameters values used in the basic fine-tuning experiment of neural machine translation model on human marking data in [KBR20]. For an extended grid search, values in **bold face** are added. 83

3.4 Variance components in *partial meta-parameter grid* search for the basic fine-tuning experiment of neural machine translation model on human marking data in [KBR20]. 86

3.5 Variance components in *full meta-parameter grid* search for the basic fine-tuning experiment of neural machine translation model on human marking data in [KBR20]. 86

3.6 Variance components in *extended meta-parameter grid* search for the basic fine-tuning experiment of neural machine translation model on human marking data in [KBR20]. 87

3.7 Variance components estimated using the minimal LMEM (3.14). 88

3.8 Variance components with nested random effect in *extended meta-parameter grid* search for neural machine translation model fine-tuned on human marking data [KBR20]. 89

4 Inferential Reproducibility 95

4.1 Text summarization results (Rouge-1/2/L) for baseline (BART) and SOTA (BART+R3F) reported in [Agh+21]. 104
4.2 Significance of result difference baseline-SOTA on CNN/DailyNews. . . . 104
4.3 Significance of baseline-SOTA on CNN/DailyNews under meta-parameter variation. 106
4.4 Variance component analysis for Rouge-1 (top), Rouge-2 (middle), and Rouge-L (bottom) estimates. 106

5 Concluding Remarks 111

Bibliography 119

Bibliography

[Agh+21] Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. “Better Fine-Tuning by Reducing Representational Collapse”. In: *International Conference on Learning Representations (ICLR)*. 2021.

[And00] Donald W.K. Andrews. “Inconsistency of the Bootstrap when a Parameter is on the Boundary of the Parameter Space”. In: *Econometrica* 68.2 (2000), pp. 399–405.

[AP08] Ron Artstein and Massimo Poesio. “Inter-Coder Agreement for Computational Linguistics”. In: *Computational Linguistics* 34.4 (2008), pp. 555–596.

[Arj+19] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. “Invariant Risk Minimization”. In: *CoRR* abs/1907.02893 (2019).

[Bar+13] Dale J. Barr, Roger Levy, Christoph Scheepers, and Harry J. Tilly. “Random effects structure for confirmatory hypothesis testing: Keep it maximal”. In: *Journal of Memory and Language* 68.3 (2013), pp. 255–278.

[Bar+20] Loıc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. “Findings of the 2020 Conference on Machine Translation (WMT20)”. In: *Proceedings of the Fifth Conference on Machine Translation (WMT)*. Online, 2020.

[Bat+15] Douglas Bates, Martin Mächler, Benjamin M. Bolker, and Steven C. Walker. “Fitting Linear Mixed-Effects Models Using lme4”. In: *Journal of Statistical Software* 67.1 (2015), pp. 1–48.

[BB12] James Bergstra and Yoshua Bengio. “Random Search for Hyper-Parameter Optimization”. In: *Journal of Machine Learning Research (JMLR)* 13 (2012), pp. 281–305.

[BBK12] Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. “An Empirical Investigation of Statistical Significance in NLP”. In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Jeju Island, Korea, 2012.

[BBL04] Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. “Introduction to Statistical Learning Theory”. In: *Advanced Lectures on Machine Learning*. Ed. by Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch. Springer, Berlin, 2004, pp. 169–207.

[BCB15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. San Diego, CA, 2015.

[BCN18] Leon Bottou, Frank E. Curtis, and Jorge Nocedal. “Optimization Methods for Large-Scale Machine Learning”. In: *SIAM Review* 60.2 (2018), pp. 223–311. DOI: 10.1137/16m1080173.

[BD21] Samuel R. Bowman and George Dahl. “What Will it Take to Fix Benchmarking in Natural Language Understanding?” In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Online: Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.naacl-main.385.

[BDB08] R Baayen, D Davidson, and D Bates. “Mixed-effects modeling with crossed random effects for subjects and items”. In: *Journal of Memory and Language* 59.4 (Nov. 2008), pp. 390–412. DOI: 10.1016/j.jml.2007.12.005.

[Ben+16] Luisa Bentivogli, Nicola Bertoldi, Mauro Cettolo, Marcello Federico, Matteo Negri, and Marco Turchi. “On the evaluation of adaptive machine translation for human post-editing”. In: *IEEE Transactions on Audio, Speech and Language Processing (TASLP)* 24.2 (2016), pp. 388–399.

[BF81] Peter J. Bickel and David A. Freedman. “Some Asymptotic Theory for the Bootstrap”. In: *The Annals of Statistics* 9.6 (1981), pp. 1196–1217.

[BG04] Yoshua Bengio and Yves Grandvalet. “No Unbiased Estimator of the Variance of K-Fold Cross-Validation”. In: *Jounal of Machine Learning Research* 5 (2004), pp. 1089–1105.

[BHT23] Stephen Bates, Trevor Hastie, and Robert Tibshirani. “Cross-Validation: What Does It Estimate and How Well Does It Do It?” In: *Journal of the American Statistical Association* (2023), pp. 1–12. DOI: 10.1080/01621459.2023.2197686.

[BMB12] Douglas Bates, Martin Maechler, and Ben Bolker. *lme4: Linear mixed-effects models using S4 classes.* R package version 1.1-5. 2012. URL: http://CRAN.R-project.org/package=lme4.

[Bon35] Carlo E Bonferroni. “Il calcolo delle assicurazioni su gruppi di teste”. In: *Studi in onore del professore salvatore ortu carboni* (1935), pp. 13–60.

[Bou+21] Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Nazanin Mohammadi Sepahvand, Edward Raff, Kanika Madan, Vikram Voleti, Samira Ebrahimi Kahou, Vincent Michalski, Tal Arbel, Chris Pal, Gael Varoquaux, and Pascal Vincent. “Accounting for variance in machine learning benchmarks”. In: *Proceedings of Machine Learning and Systems (MLSys)* 3 (2021).

[Bre01] Robert L. Brennan. *Generalizability theory.* Springer, 2001.

[Can+06] Angelo J. Canty, Anthony C. Davison, David V. Hinkley, and Valerie Ventura. “Bootstrap diagnostics and remedies”. In: *The Canadian Journal of Statistics* 34.1 (2006), pp. 5–27.

[Che11] Michael R Chernick. *Bootstrap methods: A guide for practitioners and researchers.* John Wiley & Sons, 2011.

[Cla+11] Jonathan Clark, Chris Dyer, Alon Lavie, and Noah Smith. “Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL’11).* Portland, OR, 2011.

[Coh60] Jacob Cohen. “A coefficient of agreement for nominal scales”. In: *Educational and Psychological Measurement* 20.1 (1960), pp. 37–46.

[Coh95] Paul R. Cohen. *Empirical Methods for Artificial Intelligence.* The MIT Press, 1995.

[Col17] David Colquhoun. "The reproducibility of research and the misinterpretation of p-values". In: *Royal Society Open Science* 4.12 (2017). DOI: `10.1098/rsos.171085`.

[Cro51] Lee J. Cronbach. "Coefficient Alpha and the Internal Structure of Tests". In: *Psychometrika* 16.3 (1951), pp. 297–334.

[DAm+20] Alexander D'Amour, Katherine A. Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yi-An Ma, Cory Y. McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. "Underspecification Presents Challenges for Credibility in Modern Machine Learning". In: *CoRR* abs/2011.03395 (2020). arXiv: `2011.03395`.

[Dau+14] Yann N. Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. "Identifying and Attacking the Saddle Point Problem in High-Dimensional Non-Convex Optimization". In: *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*. Montreal, Canada, 2014.

[Dav03] A. C. Davison. *Statistical Models*. Cambridge University Press, 2003.

[Dem13] Eugene Demidenko. *Mixed Models: Theory and Applications with R*. Wiley, 2013.

[Die98] Thomas G. Dietterich. "Approximate statistical tests for comparing supervised classification learning algorithms". In: *Neural Computation* 10.7 (1998), pp. 1895–1924.

[Dod+19] Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. "Show Your Work: Improved Reporting of Experimental Results". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019.

[Dro+17] Rotem Dror, Gili Baumer, Marina Bogomolov, and Roi Reichart. "Replicability Analysis for Natural Language Processing: Testing Significance with Multiple Datasets". In: *Transactions of the Association for Computational Linguistics (TACL)*. Vol. 5. 2017, pp. 471–486.

[Dro+20] Rotem Dror, Lotem Peled, Segev Shlomov, and Roi Reichart. *Statistical Significance Testing for Natural Language Processing*. Morgan & Claypool, 2020.

[Dru09] Chris Drummond. "Replicability is not Reproducibility: Nor is it Good Science". In: *Proceedings of the Evaluation Methods for Machine Learning Workshop at the 26th ICML*. Montreal, Canada, 2009.

[DS12] Morris H. DeGroot and Mark J. Schervish. *Probability and statistics*. fourth. Addison-Wesley, 2012.

[DSR19] Rotem Dror, Segev Shlomov, and Roi Reichart. "Deep Dominance - How to Properly Compare Deep Neural Models". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. Florence, Italy, 2019.

[EH16] Bradley Efron and Trevor Hastie. *Computer Age Statistical Inference. Algorithms, Evidence, and Data Science*. Cambridge University Press, 2016.

[ET93] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, 1993.

[Fer+15] Francis Ferraro, Nasrin Mostafazadeh, Ting-Hao (Kenneth) Huang, Lucy Vanderwende, Jacob Devlin, Michel Galley, and Margaret Mitchell. "A Survey of Current Datasets for Vision and Language Research". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Lisbon, Portugal: Association for Computational Linguistics, 2015. DOI: `10.18653/v1/d15-1021`.

[Fis19] Ronald A Fisher. "The correlation between relatives on the supposition of Mendelian inheritance." In: *Earth and Environmental Science Transactions of the Royal Society of Edinburgh* 52.2 (1919), pp. 399–433.

[Fis25] Ronald A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, 1925.

[Fis35] Ronald A. Fisher. *The Design of Experiments*. Hafner, 1935.

[FP19] Jessica Zosa Forde and Michela Paganini. "The Scientific Method in the Science of Machine Learning". In: *Proceedings of the ICLR 2019 Debugging Machine Learning Models Workshop.* New Orleans, LA, USA, 2019.

[Fre99] Robert M French. "Catastrophic forgetting in connectionist networks". In: *Trends in cognitive sciences* 3.4 (1999), pp. 128–135.

[FS16] Nicola Ferro and Gianmaria Silvello. "A General Linear Mixed Models Approach to Study System Component Effects". In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval.* Pisa, Italy, 2016.

[GB19] Kyle Gorman and Steven Bedrick. "We Need to Talk about Standard Splits". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL).* Florence, Italy, 2019.

[GFI16] Steven N. Goodman, Daniele Fanelli, and John P. A. Ioannidis. "What does research reproducibility mean?" In: *Sci Transl Med* 8.341 (2016), pp. 1–6.

[GL14] Andrew Gelman and Eric Loken. "The Statistical Crisis in Science". In: *American Scientist* 102.6 (2014), pp. 460–465.

[GO20] Katelyn Gao and Art B. Owen. "Estimation and Inference for Very Large Linear Mixed Effects Models". In: *Statistica Sinica* 30 (2020), pp. 1741–1771.

[Gre+14] Spence Green, Sida I. Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D. Manning. "Human Effort and Machine Learnability in Computer Aided Translation". In: *Proceedings the Conference on Empirical Methods in Natural Language Processing (EMNLP).* Doha, Qatar, 2014.

[Hag16] Michael Hagmann. "A comparison of Bayesian model selection methods for the analysis of genome wide association studies". en. In: (2016). DOI: 10.25365/THESIS.44373.

[Hal12] Kevin A. Hallgren. "Computing Inter-Rater Reliability for Observational Data: An Overview and Tutorial". In: *Tutor Quant Methods Psychol.* 8.1 (2012), pp. 23–34.

[Hal13] Peter Hall. *The bootstrap and Edgeworth expansion.* Springer Science & Business Media, 2013.

[Han06] David J. Hand. "Classifier Technology and the Illusion of Progress". In: *Statistical Science* 21.1 (2006), pp. 1–14.

[Hei+21] B.J. Heil, M.M. Hoffman, F. Markowetz, S. Lee, C.S. Greene, and S.C. Hicks. "Reproducibility standards for machine learning in the life sciences". In: *Nature Methods* 18 (2021), pp. 1122–1144.

[Hen+18] Peter Henderson, Riashat Islam, Philip Bachmann, Joelle Pineau, Doina Precup, and David Meger. "Deep Reinforcement Learning that Matters". In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*. New Orleans, LA, USA, 2018.

[Hen+59] C.R. Henderson, Oscar Kempthorne, S.R. Searle, and C.M. von Krosigk. "The Estimation of Environmental and Genetic Trends from Records Subject to Culling". In: *Biometrics* 15.2 (1959), pp. 192–218.

[Her+15] Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. "Teaching Machines to Read and Comprehend". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS)*. Montreal, Canada, 2015.

[HHL14] Frank Hutter, Holger Hoss, and Kevin Leyton-Brown. "An Efficient Approach for Assessing Hyperparameter Importance". In: *Proceedings of the 31st International Conference on Machine Learning (ICML)*. Beijing, China, 2014.

[HHS93] Peter Hall, Wolfgang Härdle, and Léopold Simar. "On the inconsistency of bootstrap distribution estimators". In: *Computational statistics & data analysis* 16.1 (1993), pp. 11–18.

[Hoe52] Wassily Hoeffding. "The Large-Sample Power of Tests Based on Permutations of Observations". In: *Annals of Mathematical Statistics* 23 (1952), pp. 169–192.

[Hol79] Sture Holm. "A Simple Sequentially Rejective Multiple Test Procedure". In: *Scandinavian Journal of Statistics* 6.2 (1979), pp. 65–70. ISSN: 03036898, 14679469. URL: `http://www.jstor.org/stable/4615733` (visited on 07/30/2023).

[Hot+05] Torsten Hothorn, Friedrich Leisch, Achim Zeileis, and Kurt Hornik. "The Design and Analysis of Benchmark Experiments". In: *Journal of Computational and Graphical Statistics* 14.3 (2005), pp. 675–699. DOI: `10.1198/106186005x59630`.

[Hot+08] Torsten Hothorn, Kurt Hornik, Mark A. van de Wiel, and Achim Zeileis. "Implementing a Class of Permutation Tests: The coin Package". In: *Journal of Statistical Software* 28.8 (2008), pp. 1–23. DOI: 10.18637/jss.v028.i08. URL: https://www.jstatsoft.org/index.php/jss/article/view/v028i08.

[HR21] Michael Hagmann and Stefan Riezler. "False perfection in machine prediction: Detecting and assessing circularity problems in machine learning". In: *In submission* (2021).

[HT09] Yosef Hochberg and Ajit C Tamhane. "Multiple comparison procedures". In: (2009).

[HTF08] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* second. New York, NY: Springer, 2008.

[Hua+20] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. "A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability". In: *Computer Science Review* 37 (2020).

[Hut18] Matthew Hutson. "Artificial intelligence faces reproducibility crisis". In: *Science* 359.6377 (2018), pp. 725–726.

[Ioa05] John P. A. Ioannidis. "Why Most Published Research Findings Are False". In: *PLOS Medicine* 2.8 (2005). DOI: 10.1371/journal.pmed.0020124.

[Jay+15] Gaya K Jayasinghe, William Webber, Mark Sanderson, Lasitha S Dharmasena, and J Shane Culpepper. "Statistical comparisons of non-deterministic IR systems using two dimensional variance". In: *Information Processing & Management* 51.5 (2015), pp. 677–694.

[Jia+19] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. "Fantastic Generalization Measures and Where to Find Them". In: *International Conference on Learning Representations (ICLR)*. Addis Ababa, Ethiopia, 2019.

[Jia18] Zhehan Jiang. "Using the Linear Mixed-Effect Model Framework to Estimate Generalizability Variance Components in R". In: *Methodology* 14.3 (2018), pp. 133–142.

[KBR20] Julia Kreutzer, Nathaniel Berger, and Stefan Riezler. "Correct Me If You Can: Learning from Error Corrections and Markings". In: *Proceedings of the 22nd Annual Conference of the European Association for Machine Translatioin (EAMT).* Lisbon, Portugal, 2020.

[Kim+19] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. "Learning Not to Learn: Training Deep Neural Networks With Biased Data". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* Long Beach, CA, USA, 2019.

[Kin+75] J. P. Kincaid, R. P. Fishburn, R. L. Rogers, and B. S. Chissom. *Derivation of new readability formulas for Navy enlisted personnel.* Tech. rep. Millington, TN: Technical Report, Naval Air Station, 1975.

[KKB20] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. "Generalization in Deep Learning". In: *CoRR* abs/1710.05468 (2020).

[KL16] Terry K. Koo and Mae Y. Li. "A Guideline of Selecting and Reporting Intraclass Correlations Coefficients for Reliability Research". In: *Journal of Chiropratic Medicine* 15 (2016), pp. 155–163.

[Kri04] Klaus Krippendorff. *Content Analysis. An Introduction to Its Methodology.* Sage, 2004.

[KSR18] Sariya Karimova, Patrick Simianer, and Stefan Riezler. "A User-Study on Online Adaptation of Neural Machine Translation to Human Post-Edits". In: *Machine Translation* 32.4 (Nov. 9, 2018), pp. 309–324. DOI: `10.1007/s10590-018-9224-8`.

[Lew+20] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL).* Online, 2020.

[LH03] Chin-Yew Lin and Eduard Hovy. "Automatic evaluation of summaries using N-gram co-occurrence statistics". In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL '03.* Edmonton, Canada:

Association for Computational Linguistics, 2003. DOI: 10.3115/1073445.1073465.

[LH23] Zhexiao Lin and Fang Han. “On the failure of the bootstrap for Chatterjee’s rank correlation”. In: *arXiv preprint arXiv:2303.14088* (2023).

[LM12] Richard J. Larsen and Morris L. Marx. *Mathematical Statistics and its Applications*. Fifth. Prentice Hall, 2012.

[LN68] Frederic M. Lord and Melvin R. Novick. *Statistical Theories of Mental Test Scores*. Addison-Wesley, 1968.

[LÖ22] A. -M. Leventi-Peetz and T. Östreich. “Deep Learning Reproducibility and Explainable AI (XAI)”. In: *CoRR* abs/2202.11452 (2022).

[Lon21] Michael A. Lones. “How to avoid machine learning pitfalls: a guide for academic researchers”. In: *CoRR* abs/2108.02497 (2021).

[LPM15] Thang Luong, Hieu Pham, and Christopher D. Manning. “Effective Approaches to Attention-based Neural Machine Translation”. In: *EMNLP*. Lisbon, Portugal, 2015.

[LS11] Ulrike von Luxburg and Bernhard Schölkopf. “Statistical Learning Theory: Models, Concepts, and Results”. In: *Handbook of the History of Logic, vol. 10: Inductive Logic*. Ed. by D. Gabbay, S. Hartmann, and J. Woods. Elsevier, 2011, pp. 651–706.

[Luc+18] Mario Lucic, Karol Kurach, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. “Are GANs Created Equal? A Large-Scale Study”. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS)*. Montréal, Canada, 2018.

[Luc+22] Ana Lucic, Maurits Bleeker, Samarth Bhargav, Jessica Forde, Koustuv Sinha, Jesse Dodge, Sasha Luccioni, and Robert Stojnic. “Towards Reproducible Machine Learning Research in Natural Language Processing”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Dublin, Ireland, 2022.

[McS+19] Blakeley B. McShane, David Gal, Andrew Gelman, Christian Robert, and Jennifer L. Tackett. “Abandon Statistical Significance”. In: *The American Statistician* 73.sup1 (2019), pp. 235–245.

[MDB18] Gabor Melis, Chris Dyer, and Phil Blunsom. "On the State of the Art of Evaluation in Neural Language Models". In: *Proceedings of the 6th Conference on Learning Representations (ICLR)*. Vancouver, BC, Canada, 2018.

[MEG76] Ruth Marcus, Peritz Eric, and K Ruben Gabriel. "On closed testing procedures with special reference to ordered analysis of variance". In: *Biometrika* 63.3 (1976), pp. 655–660.

[MFR21] Benjamin Marie, Atsushi Fujita, and Raphael Rubino. "Scientific Credibility of Machine Translation Research: A Meta-Evaluation of 769 Papers". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, 2021. DOI: `10.18653/v1/2021.acl-long.566`.

[MN89] P. McCullagh and J.A. Nelder. *Generalized Linear Models*. Second. Chapman and Hall, 1989.

[Mon17] Douglas C Montgomery. *Design and analysis of experiments*. John wiley & sons, 2017.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[MS01] Charles E. McCulloch and Shayle R. Searle. *Generalized, Linear, and Mixed Models*. Wiley, 2001.

[MW96] Kenneth O. McGraw and S. P. Wong. "Forming Inferences About Some Intraclass Correlation Coefficients". In: *Psychological Methods* 1.1 (1996), pp. 30–46.

[NB99] Claude Nadeau and Yoshua Bengio. "Inference for the Generalization Error". In: *Advances in Neural Information Processing Systems (NIPS)*. Denver, CO, USA, 1999.

[Nor89] Eric W. Noreen. *Computer Intensive Methods for Testing Hypotheses. An Introduction*. Wiley, 1989.

[NP33] J. Neyman and E. S. Pearson. "On the Problem of the Most Efficient Tests of Statistical Hypotheses". In: *Philosophical Transactions of the Royal Society of London. Series A* 231 (1933), pp. 289–337.

[Pap+02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. “BLEU: a Method for Automatic Evaluation of Machine Translation”. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*. Philadelphia, PA, 2002, pp. 311–318.

[Pau+18] Silviu Paun, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. “Comparing Bayesian Models of Annotation”. In: *Transactions of the Association for Computational Linguistics (TACL)*. Vol. 6. 2018, pp. 571–585.

[Paw01] Yudi Pawitan. *In All Likelihood. Statistical Modelling and Inference Using Likelihood.* Clarendon Press, 2001.

[PB00] José C. Pinheiro and Douglas M. Bates. *Mixed-Effects Models in S and S-PLUS.* Springer, 2000.

[PC14] Rebecca J. Passonneau and Bob Carpenter. “The Benefits of a Model of Annotation”. In: *Transactions of the Association for Computational Linguisitics (TACL)* 2 (2014), pp. 311–326.

[Pin+21] Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d’Alché-Buc, Emily Fox, and Hugo Larochelle. “Improving Reproducibility in Machine Learning Research (A Report from the NeurIPS 2019 Reproducibility Program)”. In: *Journal of Machine Learning Research (JMLR)* 22 (2021), pp. 1–20.

[Pla+19] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. “Competence-based Curriculum Learning for Neural Machine Translation”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL:HLT)*. Minneapolis, Minnesota, 2019. DOI: 10.18653/v1/N19-1119.

[Ple18] Hans E. Plesser. “Reproducability vs. Replicability: A Brief History of a Confused Terminology”. In: *Frontiers in Neuroinformatics* 11.76 (2018), pp. 1–4.

[Rei+20] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. “COMET: A Neural Framework for MT Evaluation”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online, 2020.

[RG17] Nils Reimers and Iryna Gurevych. “Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Copenhagen, Denmark, 2017.

[RG18] Nils Reimers and Iryna Gurevych. “Why Comparing Single Performance Scores Does Not Allow to Draw Conclusions About Machine Learning Approaches”. In: *CoRR* abs/1803.09578 (2018).

[RH21] Stefan Riezler and Michael Hagmann. *Validity, Reliability, and Significance: Empirical Methods for NLP and Data Science*. Springer International Publishing, 2021. DOI: 10.1007/978-3-031-02183-1.

[RK12] Stephen E. Robertson and Evangelos Kanoulas. “On Per-topic Variance in IR Evaluation”. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Portland, OR, USA, 2012.

[RM05] Stefan Riezler and John Maxwell. “On Some Pitfalls in Automatic Evaluation and Significance Testing for MT”. In: *Proceedings of the ACL-05 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*. Ann Arbor, MI, 2005.

[Sch19] Bernhard Schölkopf. “Causality for Machine Learning”. In: *CoRR* abs/1911.10500 (2019).

[SCM92] Shayle R. Searle, George Casella, and Charles E. McCulloch. *Variance Components*. Wiley, 1992.

[Sco55] William A. Scott. “Reliability of content analysis: The case of nominal scale coding”. In: *Public Opinion Quarterly* 19 (1955), pp. 321–325.

[Sel+21] Thibault Sellam, Steve Yadlowsky, Jason Wei, Naomi Saphra, Alexander D’Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, Ian Tenney, and Ellie Pavlick. “The MultiBERTs: BERT Reproductions for Robustness Analysis”. In: *CoRR* abs/2106.16163 (2021).

[SF79] Patrick E. Shrout and Joseph L. Fleiss. “Intraclass Correlations: Uses in Assessing Rater Reliability”. In: *Psychological Bulletin* 86.2 (1979), pp. 420–428.

[SGM19] Emma Strubell, Ananya Ganesh, and Andrew McCallum. “Energy and Policy Considerations for Deep Learning in NLP”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. Florence, Italy, 2019.

[Sha03] Jun Shao. *Mathematical Statistics*. Second. Springer, 2003.

[She+21] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. “Towards Out-Of-Distribution Generalization: A Survey”. In: *CoRR* abs/2108.13624 (2021).

[Sho11] Mohamed M. Shoukri. *Measures of Interobserver Agreement and Reliability*. Second. Taylor and Francis, 2011.

[SKR16] Patrick Simianer, Sariya Karimova, and Stefan Riezler. “A Post-editing Interface for Immediate Adaptation in Statistical Machine Translation”. In: *Proceedings of the Conference on Computational Linguistics: System Demonstrations (COLING Demos)*. Osaka, Japan, 2016.

[Sno+06] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. “A Study of Translation Edit Rate with Targeted Human Annotation”. In: *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA’06)*. Cambridge, MA, 2006.

[Søg+21] Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. “We Need To Talk About Random Splits”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Online, 2021.

[SW99] Helmut Strasser and Christian Weber. *On the Asymptotic Theory of Permutation Statistics*. English. WorkingPaper 27. SFB Adaptive Information Systems et al., 1999.

[Tan+20] Raphael Tang, Jaejun Lee, Ji Xin, Xinyu Liu, Yaoliang Yu, and Jimmy Lin. “Showing Your Work Doesn’t Always Work”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. Online, 2020.

[TB95] Jean Tague-Sutcliffe and James Blustein. “A statistical analysis of the TREC-3 data”. In: *NIST SPECIAL PUBLICATION SP* (1995), pp. 385–385.

[Vaa98] A. W. van der Vaart. *Asymptotic Statistics.* Cambridge University Press, 1998.

[Vap98] Vladimir N. Vapnik. *Statistical Learning Theory.* Wiley, 1998.

[VSS17] Ellen M. Voorhees, Daniel Samarov, and Ian Soboroff. "Using Replicates in Information Retrieval Evaluation". In: *ACM Transactions on Information Systems* 36.2 (2017), pp. 1–31.

[WHH05] N.A. Weiss, P.T. Holmes, and M. Hardy. *A Course in Probability.* Pearson Addison Wesley, 2005. ISBN: 9780321189547.

[Wil38] S. S. Wilks. "The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses". In: *Annals of Mathematical Statistics* 19 (1938), pp. 60–92.

[Woo03] Simon N. Wood. "Thin plate regression splines". In: *J. R. Statist. Soc. B* 65.1 (2003), pp. 95–114.

[Woo17] Simon N. Wood. *Generalized Additive Models. An Introduction with R.* second. Chapman & Hall/CRC, 2017.

[WSH06] Noreen M. Webb, Richard J. Shavelson, and Edward H. Haertel. "Reliability Coefficients and Generalizability Theory". In: *Handbook of Statistics* 26 (2006), pp. 81–214.

[WWG07] Brady T. West, Kathleen B. Welch, and Andrzej T. Galecki. *Linear Mixed Models: A Practical Guide Using Statistical Software.* Chapman & Hall/CRC, 2007.

[Yeh00] Alexander Yeh. "More accurate tests for the statistical significance of result differences". In: *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000).* Saarbrücken, 2000.

[Zha+20] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. "BERTScore: Evaluating Text Generation with BERT". In: *International Conference on Learning Representations (ICLR).* virtual, 2020.

[Zha+21] Yu Zhang, Peter Tino, Ales Leonardis, and Ke Tang. "A Survey on Neural Network Interpretability". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* (2021).

[Zho+20] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. “Extractive Summarization as Text Matching”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020. DOI: `10.18653/v1/2020.acl-main.552`.

[ZLD13] Xinshu Zhao, Jun S. Liu, and Ke Deng. “Assumptions behind Intercoder Reliability Indices”. In: *Communication Yearbook* 36 (2013), pp. 419–480.

[ZLH20] Lucas Zimmer, Marius Lindauer, and Frank Hutter. “Auto-PyTorch Tabular: Multi-Fidelity MetaLearning for Efficient and Robust AutoDL”. In: *CoRR* abs/2006.13799 (2020).